Titre: Title:	Simultaneous sequencing of incoming and outgoing semi-trailers on a cross-docking platform
Auteur: Author:	Mohammad Yousef Maknoon
Date:	2007
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Maknoon, M. Y. (2007). Simultaneous sequencing of incoming and outgoing semi-trailers on a cross-docking platform [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8017/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/8017/
Directeurs de recherche: Advisors:	
Programme: Program:	Unspecified

UNIVERSITÉ DE MONTRÉAL

SIMULTANEOUS SEQUENCING OF INCOMING AND OUTGOING SEMI-TRAILERS ON A CROSS-DOCKING PLATFORM

MOHAMMAD YOUSEF MAKNOON
DÉPARTEMENT DE MATHÉMATIQUES
ET DE GÉNIE INDUSTREL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INDUSTRIEL)
AOÛT 2007

Your file Votre référence
ISBN: 978-0-494-35690-6
Our file Notre référence
ISBN: 978-0-494-35690-6

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

SIMULTANEOUS SEQUENCING OF INCOMING AND OUTGOING SEMI-TRAILERS ON A CROSS-DOCKING PLATFORM

présenté par : MAKNOON, Mohammad Yousef
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. AGARD Bruno, Doctorat, président
M. BAPTISTE Pierre, ing., Doct., membre et directeur de recherche
M. SOUMIS François, Ph.D., membre

To my dear parents

and my beloved Shadi

for their unconditional love and support

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Professor Pierre Baptiste for accepting me as his student, for his confidence in my abilities and for his encouragements during my work which made difficulties much easier during the last one year.

I would like to thank my father, mother and my beautiful wife for their patience, unconditional love, encouragements and supports. This work was not possible without their presence.

Also I would like to thank my father and mother in-law, Dr. Rafii's family for their supports during these two years which makes our life happier.

RÉSUMÉ

Ce travail consiste à étudier le problème d'ordonnancement dans une plateforme de crossdocking. Une plateforme de crossdocking ressemble à un entrepôt avec des portes de chargement et des portes de déchargement. Les produits reçus de différents fournisseurs sont déchargés à leurs arrivées, consolidés et chargés dans des camions en partance à leurs destinations finales. Les produits déchargés sont soit directement chargés dans le camion en partance si ce dernier est présent à quai, soit ils sont temporairement stockés en attendant le camion adéquat. L'augmentation du nombre de produits directement chargés mène à une réduction des niveaux de stocks et à une utilisation plus efficace de la plateforme. Dans ce travail, l'ordonnancement simultané des semi-remorques à l'arrivée et en partance du crossdock est étudié pour maximiser les chargements directs des produits entre les portes de déchargement et de chargement.

Dans cette étude, on s'est limité au cas d'une plateforme à une seule porte de déchargement et une seule porte de chargement. D'abord, nous avons modélisé le problème en programme linéaire en nombres entiers. Ce programme résout le problème pour un nombre restreint de produits dans un temps raisonnable. Cependant, à mesure que le nombre de produits augmente, le temps de calcul devient grand et le programme n'est donc pas performant. Par conséquent, nous avons proposé quelques heuristiques pour résoudre le problème. Trois problèmes d'ordonnancement sont étudiés. Les cas diffèrent par la connaissance ou non des séquences d'entrée et de sortie des semi-remorques. Pour les résoudre, nous avons développé un programme dynamique et différentes heuristiques. Les résultats indiquent que l'ordonnancement des semi-remorques à l'arrivée et en partance du crossdock augmente de façon considérable la proportion des produits directement chargés par rapport au nombre total des produits chargés(directement ou pas), qui mène à une réduction du niveau de stock. Nous concluons qu'un bon ordonnancement des semi-remorques améliore la performance interne des plateformes de crossdocking.

ABSTRACT

This study presents the semi-trailer scheduling problem on a cross-docking platform. There are sequences of inbound and outbound semi-trailers on the transshipment platform. At the inbound door, the products are unloaded, broken down, processed and consolidated for reshipment at the outbound door. The unloaded products transfer directly to the outbound door if they move to the currently loading semi-trailer; otherwise, the products are put into temporary storage for future reshipment. Increasing the proportion of directly transiting products leads to a reduction in inventory levels and a more efficient utilization of platform transporters. In this work, the simultaneous sequencing of inbound and outbound semi-trailers is studied to maximize the direct flow of products between receiving and shipping doors.

In this study, the problem is limited to one receiving and one shipping door. First we present the mathematical formulation for the problem. This formulation solves the problem for a small number of products in a reasonable time. However, as the number of products increases, the formulation becomes impractical. Therefore, we propose some heuristic methods to solve the problem.

Three cases of the sequencing problem in transshipment are studied. The cases differ by the available knowledge about the incoming or outgoing sequences. Dynamic programming and heuristics are used. These general methods are integrated with Tabu search as a resolution for the cases. The results indicate that sequencing incoming and outgoing semi-trailers notably increases the ratio of directly transiting products to total transferred products, which leads to a reduction in inventory level and transporter utilizations. We conclude that sequencing semi-trailers is one of the platform operational activities that can increase cross-docking performance.

CONDENSÉ EN FRANÇAIS

La chaîne d'approvisionnement est un système coordonné pour déplacer les produits du fournisseur au client. Le but de la chaîne d'approvisionnement est de diminuer le coût en coordonnant les activités et l'information entre ses différentes parties. Le coût logistique se compose du coût de transport, de celui de l'entreposage, et constitue l'un des coûts non profitables pour le système. Par conséquent, la réduction de ce coût constitue l'un des objectifs de la chaîne d'approvisionnement.

Traditionnellement, les activités d'entrepôt consistent en la réception, le stockage et l'expédition des produits. Comme le stockage et le chargement des commandes s'avèrent être les plus coûteux de ses activités, le transbordement avec son caractère spécifique de l'environnement "juste à temps" et un moindre stockage peut être considéré comme une solution attractive. Kinnear [1] a défini le système de "Cross-dock" comme une plateforme de transbordement qui reçoit le produit d'un fournisseur pour plusieurs destinations et le consolide avec d'autres produits pour une destination finale commune de livraison.

Dans la plateforme de transbordement, la gestion des activités opérationnelles est un problème important qui affecte son efficacité. Elle a besoin de l'infrastructure appropriée et du programme d'ordonnancement pour pouvoir coordonner les activités. L'ordonnancement de semi-remorques entrantes et sortantes est l'un des problèmes d'ordonnancement qui joue un rôle important dans l'efficacité de plateforme.

Il existe différents genres de plateforme de transbordement basés sur leur rôle dans le système. En d'autres termes, la plateforme de transbordement peut recevoir les produits d'un fournisseur et les distribuer entre différents consommateurs ou recevoir des produits de plusieurs fournisseurs pour un seul consommateur ou encore elle pourrait relier plusieurs fournisseurs et consommateurs. Dans cette recherche nous ne considérons que le dernier cas, en l'occurrence, la plateforme de transbordement « de correspondance multivoque ».

La plateforme de transbordement se compose de portes de réception, d'inventaire provisoire et de portes d'expédition. À la porte d'arrivée les produits, différents selon leurs destinations d'envoi, sont déchargés. Puis, ils sont décomposés, traités et consolidés pour la réexpédition à la porte d'expédition. Les produits déchargés peuvent se déplacer de la façon suivante :

- Transférer les produits directement à partir de la porte de réception jusqu'à à la porte d'expédition.

- Mettre les produits dans l'entrepôt de la plateforme en attendant de les transférer à leur destination finale (habituellement moins de 24 heures).

La deuxième manière de transférer augmente le niveau de stock, en plus d'occuper les transporteurs de plateforme. L'efficacité est définie comme le rapport entre le nombre de produits directement transférés et le nombre total de tous les produits transférés dans un temps donné.

Dans la pratique, la plateforme a plusieurs arrivées et plusieurs sorties. Dans cette étude, nous nous limiterons à une seule arrivée et une seule sortie. Cette restriction n'est pas réaliste, mais elle peut être employée comme situation de base pour d'autres dispositions. Dans ce modèle, les hypothèses suivantes sont assumées :

- Chaque semi-remorque quitte la porte d'arrivée quand il est complètement déchargé. De l'autre côté, chaque semi-remorque quitte la porte quand elle est complètement chargée.
- Les opérations internes du transbordement, tels que le triage et la mise en bloc, ne sont pas considérées.
- La capacité de stockage est illimitée.
- Chaque semi-remorque quitte la plateforme vers une seule destination.

- Toutes les semi-remorques entrantes et sortantes sont disponibles au temps zéro.
- Le nombre de produits à l'arrivée et en partance sont égaux.
- Les produits diffèrent par leur destination.
- Le chargement, le temps de déchargement et le temps de transfert sont constants et ne sont pas considérés.
- Le temps considéré pour le problème peut être un quart de travail ou un jour.
- Le nombre et la capacité de semi-remorques entrantes et sortantes sont égaux.
- Il n'y a aucune règle pour décharger des produits de semi-remorque.

Le problème est illustré sur le schéma 1 :

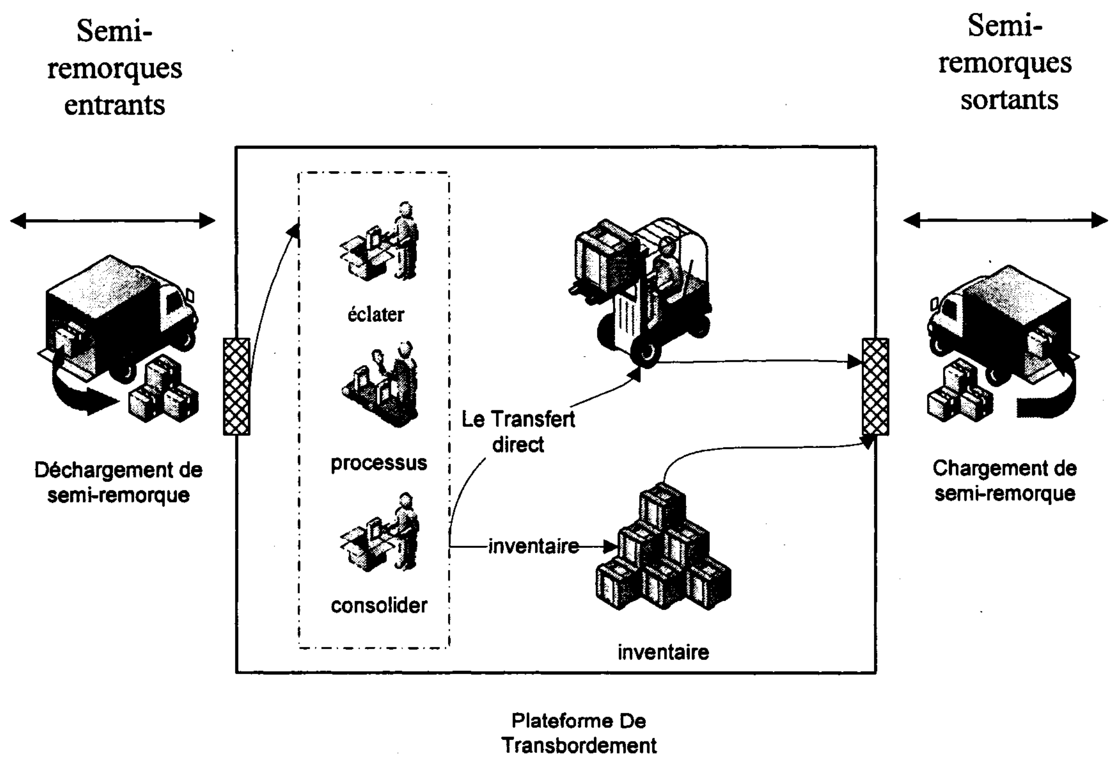

Schéma 1 : Description du problème

Dans la littérature, l'efficacité de transbordement est étudiée au niveau global et au niveau de la plateforme. Au niveau global, c'est le réseau de plateforme qui est considéré. L'étude des réseaux vise principalement à réduire le nombre de semi-remorques et le niveau de stock. Au niveau de la plateforme, les études sont concentrées sur les activités opérationnelles et sur la conception de la plateforme.

Le nombre de plateformes jouent un rôle important dans l'efficacité globale de la chaîne d'approvisionnement. Zhang M. [2] a défini deux types de stratégies d'expéditions. La première est d'expédier quand les semi-remorques sont pleines. On l'appellera expédition à chargement complet. La seconde est d'expédier les semi-remorques à temps fixe, même si elles ne sont pas complètement remplies. On l'appellera expédition à temps fixe. Ratliff et al. [3] ont étudié un réseau de plateforme à chargement complet dans l'industrie de l'automobile. Ils ont obtenu le nombre de chaque plateforme et l'écoulement d'expédition entre eux. Afin de réduire l'inventaire, ils ont réduit au minimum le nombre de semi-remorques dans la plateforme. La même étude a été faite dans un réseau de plateforme à temps fixe par Donaldson et al. [4] pour le service postal des États-Unis.

Ping Chen et al. [5] a élargi les problèmes ci-dessus pour réduire au minimum le coût de transport et le niveau d'inventaire en considérant la fenêtre de temps du chargement et du déchargement, la capacité de stockage de plateforme et le niveau global d'inventaire.

Dans Lee et al. [6], les auteurs réduisent le niveau des stocks et le délai d'expédition dans le réseau de plateforme en intégrant un modèle de transbordement et d'ordonnancement de semi-remorques. Dans leur modèle intégré, tous les produits se sont déplacés du fournisseur au client sans aucune perte de temps inutile.

Au niveau global, on n'assigne pas les semi-remorques à des portes particulières, et on n'étudie pas la circulation des produits à l'intérieur de chacune des plateformes. Au niveau de plateforme, pour augmenter l'efficacité, on considère principalement le problème de circulation des produits. Dans la littérature, la conception de la plateforme et la gestion des activités opérationnelles sont les principaux facteurs étudiés pour contrôler la circulation des produits. Gue et Kang [7] ont simulé une plateforme et déclarent que le système à deux étages (chargement et déchargement) a un rendement inférieur à un système à un seul étage. Dans Bartholdi et Gue [8], les auteurs réduisent au minimum le coût de la main-d'œuvre en développant des modèles qui tiennent compte du coût de transport entre les portes et les congestions qui se produisent pendant la mise en bloc. Dans [9], ils ont démontré que les modèles d'acheminement de produits sont déterminés par la conception de la plateforme, la géométrie, le mouvement de matériel, et l'ordonnancement.

Schaffer [10] a déclaré que la gestion opérationnelle est l'une des conditions pour l'implantation réussie de transbordement. Le premier travail sur des activités opérationnelles de plateforme a été effectué par Tsui et Chang [11] et ils ont formulé l'attribution des portes comme modèle de programmation en nombre entier. D'ailleurs, ils proposent un outil d'aide à la décision utilisent micro-ordinateur pour assigner des portes de dock [12].

Li et al. [13] étudient le problème de chargement et déchargement d'ordonnancement dans la plateforme de transbordement. Chaque semi-remorque doit quitter la plateforme dans un temps prédéterminé. Ils ont employé le problème d'ordonnancement de machine comme idée pour modeler le problème. Dans leur modèle la plateforme de transbordement est divisée en secteurs de chargement et de déchargement. Les dates d'arrivée des semi-remorques entrantes sont variables. Les articles reçus sont embarqués directement ou envoyés au secteur d'exportation afin d'être chargés pour la réexpédition. Dans leur problème, le temps pour commencer à décharger les semi-remorques est programmé pour que chaque chargement de semi-remorques soit accompli au temps prévu.

L'opération de transfert de plateforme dans l'industrie de colis est étudiée par D.L. Mc Williams et al. [14]. La plateforme étudie consiste en une porte d'arrivée, une porte de partance et d'un convoyeur. Ils réduisent au minimum l'intervalle de temps entre le premier colis déchargé et dernier colis chargé.

Dans Yu W. et al. [15], les auteurs ont minimisé le temps d'exécution du chargement et du déchargement, en faisant l'ordonnancement des semi-remorques, quand la marchandise est aux portes de déchargement. Ils ont proposé une formulation mathématique pour les petits échantillons et une heuristique pour des plus gros échantillons.

L'ordonnancement des semi-remorques est l'une des opérations les plus importantes dans le transbordement. Baptiste et al. [16] classifient les problèmes d'ordonnancement de semi-remorques avec les critères suivants :

- La connaissance ou non de l'ordre d'arrivée et de partance
- Le nombre de semi-remorques pour chaque destination (un ou plusieurs)
- Le modèle des files d'attente (FIFO, Libre)

Ils proposent les quatre sous problèmes qui sont solubles de façon polynômiale.

Dans cette étude, nous nous penchons sur le problème d'ordonnancement des semi-remorques. Nous considérons le nombre de produits transférés directement comme critère d'évaluation.

D'abord nous formulons le problème avec des variables binaires. Dans la formulation du problème, les semi-remorques entrantes et sortantes et les chargements et les déchargements des produits sont ordonnés dans le but de maximiser le nombre de produits qui transfèrent directement. Le modèle performe en temps raisonnable pour des petits problèmes, mais il est impraticable pour des problèmes moyens et grands. Pour cette raison, nous avons proposé quelques algorithmes heuristiques qui fonctionnent dans un temps raisonnable. Pour étudier le problème plus en détail, nous avons classifié notre problème en trois cas qui diffèrent par la connaissance des ordres entrants et sortants.

Dans le modèle mathématique, la solution optimale est affectée par les variables de décision suivantes:

1. Ordre entrant de semi-remorque
2. Ordre sortant de semi-remorque
3. Ordre de déchargement des produits
4. Politique du déchargement

Les première et deuxième variables sont des ordres de semi-remorque. Pour la troisième variable, chaque semi-remorque entrante contient les articles à transporter à différentes destinations. Les articles qui doivent être embarqués dans une semi-remorque sortante doivent être déchargés d'abord. La troisième variable indique l'ordre de déchargement des produits. Cette variable peut être une constante et est due aux limites techniques des opérations de déchargement. Par exemple, si les produits déchargent dans l'ordre de FIFO ou de LIFO, la troisième variable est constante, autrement elle est déterminée par le modèle. Pour la quatrième variable, supposez la situation suivante: une remorque sortante est placée à la porte en partance et des articles attendent déjà pour être embarqués vers leur destination. Le directeur peut choisir d'embarquer ces articles ou d'attendre jusqu'à ce qu'une semi-remorque entrante arrive avec des articles qui peuvent être embarqués directement à la destination. Par conséquent, il y a deux politiques possibles extrêmes : ou les produits de stockage sont systématiquement employés pour compléter une semi-remorque (peu d'inventaire), ou des articles de stockage sont embarqués dans la dernière semi-remorque allant à la destination. La politique optimale est une combinaison de ces deux extrêmes.

Vu ces prétentions, on propose trois cas :

- **Cas1 :** Les ordres de semi-remorques entrantes et sortantes sont connus et les variables 3 et 4 sont examinées.

- **Cas 2 :** L'ordre de semi-remorques entrantes est connu et les variables 2, 3 et 4 sont examinées.
- **Cas 3 :** Aucune information sur des ordres de semi-remorques n'est connue, et les variables 1, 2, 3 et 4 sont examinées.

Pour le premier cas du problème nous considérons l'énumération complète des possibilités. Pour ce faire, nous employons un algorithme basé sur la programmation dynamique et un graphique pour le présenter. L'algorithme proposé nous a donné la solution optimale. Il est présenté comme suit :

L'algorithme de chargement et déchargement (LUA) :

Indice:

z : Nombre total de semi-remorques entrantes

j : Destination de chargement

l : Nombre total de semi-remorques sortantes

N : Nombre d'identification du nœud

AD: Destination courante de chargement

d : Destinations

TC : capacité de la semi-remorque

P_d^k : Nombre de produits pour la destination d dans la semi-remorque k

Les éléments d'un nœud S:

S_j^N=(H_d,G_d,c,k)

S_j^N : Nœud N dans semi-remorques sortantes j

H_d : Le vecteur contenant les variables indiquant le nombre possible de produits qui sont transférés directement pour chaque destination

G_d: Le vecteur contenant les variables indiquant le nombre de produits dans l'inventaire temporaire pour chaque destination

c : Coût (nombre total de produits qui ont été transférés directement du début jusqu'au nœud courant)

k : numéro dans l'ordre de la semi-remorque déchargée

Début

Initialiser : H_d=P_d^1, G_d=0, C=0, k=1 (∀d) (1)

Créer nœud: S_0^0 = (H_d,G_d,c,k) (2)

Pour j=0 à l (3)

  Pour tous les nœuds dans j (4)

    Faire jusqu'à H_{AD} >= TC+1 (5)

      Si H_{AD}>=TC (6)

        H_{AD}=H_{AD}-TC, c= c+TC (7)

        Créer nœud S_K^N = (H_d,G_d,c,k) (8)

        Restituer H_d et c

      Sinon si H_{AD}+G_{AD}>= TC (9)

        H_{AD}=0, G_{AD} = G_{AD} - (TC-H_{AD}), (10)

$c=c+H_{AD}$

Créer nœud $S_K^N = (H_d, G_d, c, k)$ (11)

Restituer H_d,G_d et c

Si k<=z (12)

$H_{AD}=H_{AD} + P_{AD}^K$, $G_d=G_d+ H_d$ $(\forall d \neq AD)$,k=k+1 (13)

Autrement sortir de la boucle (14)

Appeler la fonction de dominance (15)

Fin

Cet algorithme commence par un nœud initial (1) et (2). Puis, il considère tous les nœuds produits pour la semi-remorque précédente en chargement (3) et (4), et pour chaque nœud, il produit tous les prochains nœuds pour la semi-remorque actuellement en chargement (5) à (14). De (6) à (8) l'algorithme considère que tous les produits chargés sont directement transférés. De (9) à (11) l'algorithme considère des produits de stockage en plus des produits directement chargés. De (12) à (14) l'algorithme permet de décharger la prochaine semi-remorque entrante. À fin de l'algorithme, le nœud de la dernière semi-remorque chargée possédant le coût le plus élevé est choisi comme le nœud final. Le chemin parcouru, du nœud initial jusqu'au nœud final choisi, est la politique optimale pour le chargement/déchargement.

Dans la pratique, deux règles de domination (15) sont employées pour omettre des nœuds inutiles.

Fonction de dominance:

$S_K^p=(H_d(p), G_d(p), c(p), k(p))$ dominate $S_K^q=(H_d(q), G_d(q), c(q), k(q))$ si:

1) $k(p) < k(q), H_d(p) = H_d(q) \quad \forall d, and\ c(p) \geq c(q)$

2) $k(p) = k(q), c(p) = c(q)\ and \sum_d H_d(q) \leq \sum_d H_d(p)$

La première règle stipule que, si pour chaque destination, les produits transférés directement pour deux nœuds sont égaux ; le nœud avec le coût le plus élevé domine l'autre. La deuxième règle s'applique dans le cas ou le cout est le même pour deux ou plusieurs nœuds. Dans ce cas, le nœud possédant le nombre le plus élevée de produits transférés directement pour toutes les destinations domine les autres.

Pour la résolution de notre deuxième cas, on a adopté deux approches : la première consiste à introduire la recherche Tabou dans l'algorithme du premier cas, et plus précisément pour les semi-remorques sortantes afin d'améliorer leur l'ordonnancement, ce qui permet d'augmenter le nombre de produits transférés directement. Pour atteindre les mêmes objectifs, la deuxième approche utilise une méthode heuristique. Dans ce qui suit, on présente en détail ces deux algorithmes.

L'algorithme d'ordonnancement sortant (SOA) :

Étape 1 : Exécuter l'algorithme de chargement/déchargement pour la valeur initiale.
Étape 2 : Choisir deux numéros de semi-remorque.

Étape 3 : Permuter ces numéros et les enregistrer dans la liste Tabou (s'ils ne vont pas à la même destination et ne sont pas déjà dans la liste Tabou).

Étape 4 : exécuter l'algorithme de chargement/déchargement.

Étape 5 : Enregistrer le coût et, s'il est amélioré, enregistrer le chemin optimal.

Étape 6 : Passer à l'étape 2 ou arrêter si le coût n'est pas modifié après 20 itérations.

L'algorithme glouton d'ordonnancement sortant:

Étape 1 : Créer une valeur initiale (le nombre de produits transférés directement pour chaque destination est égale aux nombre de produits déchargés de la première semi-remorque ; le reste est mis à zéro).

Étape 2 : Pour toutes les semi-remorques chargées faire les étapes 3 à 8.

Étape 3 : Pour toutes les destinations faire les étapes 4 à 7.

Étape 4 : Répéter les étapes 5 et 8 aussi longtemps que le total des produits transférés directement pour la destination choisie est plus grand que la capacité de la semi-remorque.

Étape 5 : Calculer le coût pour la destination courante choisie; préserver les résultats si il y a amélioration.

Étape 6 : Considérer la prochaine semi-remorque déchargée, mettre à jour les valeurs et passer à l'étape 4.

Étape 7 : sauvegarder la meilleure affectation.

Étape 8 : Sélectionner la meilleure affectation sortante en tant que affectation courante puis passer à l'étape 2.

Étape 9 : La liste finale est l'affectation sortante optimale

Pour résoudre le troisième cas, on utilise les deux algorithmes du deuxième cas en ajoutant une recherche Tabou pour les semi-remorques entrants afin d'améliorer leur l'ordonnancement, ce qui permet d'augmenter le nombre de produits transférés directement.

Les structures des deux algorithmes sont pratiquement identiques avec quelques différences dans les fonctions appelées. Ils sont décrient ci-dessous. Les étapes du premier algorithme SIOA (Sequencing incoming/outgoing algorithm) sont présentées.

Dans le cas ou le deuxième algorithme SIOGA (Sequencing incoming/outgoing greedy algorithm) est différent de SIOA l'étape est décrite entre parenthèses.

L'algorithme d'ordonnancement entrant-sortant (SIOA) / L'algorithme glouton d'ordonnancement entrant-sortant (SIOGA):

Étape 1 : Exécuter l'algorithme d'ordonnancement sortant (ou l'algorithme glouton d'ordonnancement sortant) pour la valeur initiale.

Étape 2 : Choisir deux numéros de semi-remorque déchargée.

Étape 3 : S'ils ne sont pas dans la liste Tabou, permutez les numéros choisis et les enregistrer dans la liste Tabou.

Étape 4 : Exécuter l'algorithme d'ordonnancement sortant (ou l'algorithme glouton d'ordonnancement sortant)

Étape 5 : Enregistrer le coût et, s'il augmente, enregistrer le chemin optimal.

Étape 6 : Passer à l'étape 2 ou arrêter si le coût n'est pas amélioré après 20 itérations.

Pour tester nos algorithmes, nous les avons appliqués sur des échantillons de petite, moyenne et grande taille. Nous avons fait l'énumération complète pour le problème de petite taille afin d'étudier les écarts entre les résultats obtenus par tous les algorithmes proposés et la solution optimale. Les résultats montrent que les algorithmes basés sur le premier cas sont plus performants que les algorithmes basés sur la méthode glouton, mais l'écart entre eux est négligeable. Pour le troisieme cas, on observe que les deux algorithmes présentent de bons résultats (l'écart est inferieur à 2%).

Finalement, nous avons constaté que l'ordonnancement des semi-remorques qui entrent et sortent augmente la performance de plateforme.

TABLE OF CONTENTS

DEDICATION iv
ACKNOWLEDGEMENTS v
RÉSUMÉ vi
ABSTRACT vii
CONDENSÉ EN FRANÇAIS viii
TABLE OF CONTENTS xxi
LIST OF TABLES xxii
LIST OF FIGURES xxiii
LIST OF APPENDICES xxiv
CHAPTER 1: INTRODUCTION 1
CHAPTER 2: STUDIES ON CROSS-DOCKING 4
2.1 Cross-docking networks 5
2.2 Platform performance 8
CHAPTER 3: MODEL DESCRIPTION AND MATHEMATICAL FORMULATIONS 12
3.1 Model description 12
3.2 Mathematical formulation 14
CHAPTER 4: RESOLUTION APPROACHES 17
4.1 First case resolution approaches 20
4.2 Second case resolution approaches 24
4.3 Third Case resolution approaches 26
CHAPTER 5: EXPERIMENTS 27
5.1 Data 27
5.2 Model 34
CHAPTER 6: CONCLUSION AND DISCUSSION 37
REFFERENCES 38
APPENDICES 40

LIST OF TABLES

Table 2-1: Research at the global level 7
Table 2-2: Research at platform level 10
Table 4-1: Example 23
Table 5-1: Distribution of semi-trailers for each case 28
Table 5-2: The Best and The worst bound 28
Table 5-3: Average Number of direct transiting products (Small problem)............... 29
Table 5-4: Algorithm performance.. 29
Table 5-5: Algorithm improvements for five, ten and twenty semi-trailers................ 33
Table 5-6: Number of generated nodes for different numbers of products............... 34
Table 5-7: Generated nodes with different numbers of destinations 35

LIST OF FIGURES

Figure 1-1: Model description 12
Figure 4-1: Loading/unloading policy 18
Figure 4-2: Optimal policy algorithm for the example 24
Figure 5-1: Comparing the algorithms' performance for second case with optimal solution (case 2-2-1 data)... 30
Figure 5-2: Comparing algorithms' performance for third case with optimal solution (case 2-2-1 data) 31
Figure 5-3: Compare algorithms performance for three destinations 32
Figure 5-4: Compare algorithms performance for five destinations 32
Figure 5-5: Compare algorithms performance for ten destination 32
Figure 5-6: Number of generated nodes for different numbers of products 35
Figure 5-7: Average number of generated nodes for different destinations 36
Figure 5-8: The standard deviation of the average generated nodes for different destinations 36

LIST OF APPENDICES

APPENDIX A :

SIMULTANEOUS SEQUENCING OF INCOMING AND OUTGOING SEMI-TRAILERS ON A CROSS-DOCKING PLATFORM 40

CHAPTER 1: INTRODUCTION

A supply chain is a coordinated system for moving products from supplier to customer. Its aim is to decrease costs by coordinating activities and information between its parts and moving to a just-in-time environment. Logistic costs, which consist of transportation and warehousing cost, is non-beneficial cost for the system. Therefore, reducing the logistic costs is one of the goals of the supply chain.

Traditional warehouse activities consist of the receiving, storing and shipping of products. Storing and picking up orders are costly, making cross-docking, with a just-in-time environment and low inventory, is an attractive solution. Kinnear [1] defined the cross-docking system as a transshipment platform where products are received from a supplier for several destinations and consolidated with other suppliers' products for a common final delivery destination.

On a transshipment platform, managing operational activities is an important issue that affects performance. A suitable infrastructure and scheduling system are required to coordinate activities. The sequencing of incoming and outgoing semi-trailers is a scheduling problem that plays an important role in platform performance.

There are different kinds of cross-docking, based on its role in the system. It can receive products from one supplier and distribute them to several consumers, receive products from many suppliers for one consumer, or connect many suppliers to many consumers. This research focuses on the third type of cross-docking system.

A cross-docking platform consists of inbound doors, temporary storage and outbound doors. At inbound doors, the incoming products, which differ by their places of destination, are unloaded. They are classified, processed and consolidated for reshipment at outbound doors. The unloaded products can move in one of the following manners:

- Products can be transferred directly from the inbound to the outbound door.
- Products can be transferred to temporary storage to wait for future reshipment to their final destination (usually less than 24 hours)

Using temporary storage increases inventory levels and occupies the platform transporters. In addition, if the products have a due date for reshipment, a fine may be incurred if storing them causes a delay. In this discussion, performance is defined as the ratio of directly transferred products to all products transferred in a defined unit of time.

This study investigates the sequencing problem on a transshipment platform. The objective is to maximize the number of directly transiting products. First we present the mathematical formulation for the problem. This formulation solves the problem for a small number of products in a reasonable time. However, as the number of products increases, the formula becomes impractical. Therefore, we propose some heuristic methods to solve the problem.

We identify four types of decision variable that influence the number of directly transiting products: incoming sequence, outgoing sequence, unloading order of products, and loading and unloading policy. We propose three cases for the sequencing problem:

1. Both sequences are known and we identify the values of the third and fourth variables.

2. Incoming sequences are known and the remaining variables are obtained.

3. No information is known about the decision variables.

For the first case, we consider the complete enumeration of possibilities using an algorithm based on dynamic programming. The proposed algorithm gives us the optimal solution. For the second case, we develop the algorithm with Tabu search to obtain an optimal outgoing order of products, and we propose a heuristic method based on a greedy algorithm to obtain the outgoing sequence. For the third case, the algorithms in the second case are integrated with evolutionary search to obtain the best incoming and outgoing sequences.

We tested the algorithm using small, medium and large sample sizes. We used the complete enumeration method for the small sample to investigate the gap between our proposed algorithm and the optimal solution. The results indicate that algorithms based on the first case perform better than algorithms based on the greedy method, but the gap between them is negligible. For the small sample, we conclude that both algorithms obtained good results (the gap is less than 2%). Finally, we conclude that sequencing incoming and outgoing semi-trailers increases platform performance.

CHAPTER 2: STUDIES ON CROSS-DOCKING

One of the issues in supply chain management is controlling the physical flow of products. This is a challenging activity that can decrease costs and increase customer satisfaction. The use of cross-docking centers has been proposed as a way to manage product flow more efficiently in a supply chain.

In this section, we review the current research on cross-docking. Cross-docking platforms are generally studied at two levels. At the global level, studies focus on the cross-docking network. The research at this level has mainly been aimed at finding the ideal number and location of platforms to reduce the number of semi-trailers and control product flow. Section 2.1 is dedicated to this subject.

At the platform level, product flow and timing are the main variables seen to contribute to increase platform efficiency. Most researches at this level have focused on controlling product flow through the layout of the platform and the management of operational activities. In section 2.2, we survey the research that has been done at the platform level.

2.1 Cross-docking networks

There are two types of cross-docking network, which differ according to their dispatching strategy. Zhang M. [2] defined two strategies for dispatching semi-trailers: load-driven and schedule-driven.

In load-driven systems, the semi-trailers dispatch when there are sufficient products on the platform. In this system, the main focus is on transporter utilization, rather than on service.

The load-driven network for the U.S. automobile industry was studied by Ratliff et al. [3]. The distribution network consists of plants, mixing centers and local distribution centers (ramps). The objective of the study was to minimize the average delay between production and delivery, which was caused by transporting duration and wait time (how long the products wait in the mixing center before being loaded). The number and location of mixing centers and the routing of products were determined. In addition, there were decisions as to whether to transfer the automobiles directly from plants to ramps or whether to use mixing centers. In the system, there were two approaches for sending the products from plants to local distribution centers. In the first approach, the products for different ramps were consolidated and shipped from plants to a single mixing center. In the second approach, the products were consolidated through mixing centers. Mix integer linear programming was used to determine which method worked best and how products should be routed in the network.

In a schedule-driven network, service is more important than vehicle utilization. In this dispatching strategy, each vehicle departs the platform according to a predefined schedule. Donaldson et al. [4] investigated the schedule-driven network used by the United States Postal Service. The network consists of sending centers, cross-docking platforms and receiving centers. In each center, parcels are received, processed and reshipped to other centers. The objective was to minimize the number of semi-trailer miles to satisfy transport requirements with respect to the delivery due date for each parcel. In the system, each parcel could be transferred by semi-trailer or by airplane. The preferred method was to use semi-trailers; however, if the parcel could not be delivered on its due date by a semi-trailer, it would be transferred by air. Integer programming was used to model the transportation cost.

Chen P. et al. [5] generalized from the previous research to minimize global transportation costs by considering pickup and delivery time windows, platform storage capacity and inventory. The objective was to minimize global costs, including those associated with transportation and inventory. Two heuristic methods (integrated with simulated annealing and Tabu search) were developed as resolution approaches.

Lee et al. [6] reduced inventory level and delivery lead-time in a cross-docking network by integrating the transshipment model and the scheduling of semi-trailers. They considered how long each semi-trailer had to wait to be loaded and how products were consolidated for each destination. They proposed that simultaneous arrival and consolidation caused the products move from supplier to customer without interruption. To achieve this, the network was broken down into three sections: the network from the supplier to the cross-dock, the cross-dock, and the network from the cross-dock to the customer. They integrated vehicle routing and scheduling with cross-docking consolidation to decrease inventory level and delivery lead-time. The problem was NP-hard and a Tabu search heuristic algorithm was proposed as a resolution approach.

Overall, the main goals at the global level are to design networks and control the flow of products from supplier to customer and to schedule semi-trailers to decrease the transfer time (except in a schedule-driven network; see Table 2–1). At this level, inventory and semi-trailers utilization are the two factors that have been considered by previous research.

Table 2-1: Research at the global level

Global level				
Work	[3]	[4]	[5]	[6]
Objective	Minimize the time between production and delivery	Minimize the number of truck miles to satisfy the transport requirement	Minimize global transportation cost (transportation + delivery)	Reduce inventory and lead time by integrating products' arrival and consolidation
Considered factor				
Timing	☒	☐	☒	☒
Product flow	☒	☒	☒	☒
Inventory	☐	☐	☒	☒
Transporters (Semi-trailers)	☐	☒	☒	☐

2.2 Platform performance

At the platform level, the flow of products is the main variable considered to increase platform efficiency. Designing the platform layout and managing operational activities have been studied as methods to control product flow.

There is some research indicating that platform efficiency depends on the physical layout of the system. Gue and Kang [7] simulated the platform and concluded that a two-stage system (loading and unloading) has lower output than a single-stage system when the freights are blocked between stages.

Gue and Bartholdi [8][9] studied the pattern of doors on a transshipment platform. They stated that determining the door pattern is an inexpensive way to reduce labor cost. They developed models for the travel cost between the doors and the congestion that occurs during consolidation. In their model, freight flow patterns are determined by platform layout and geometry, material handling systems, freight mix and scheduling. To solve the model, they used a simulated annealing procedure to interchange pairs of semi-trailers, and then a cost model to evaluate the results [8][9].

Schaffer [10] suggested that operational management is a requirement for successful cross-docking platform. The first work on platform operational activities was done by Tsui and Chang [11], who formulated the dock assignment problem—simultaneously allocating both inbound and outbound doors to semi-trailers—as an integer programming model. They proposed a microcomputer-based decision support tool to assign dock doors in a freight yard [12].

Li et al. [13] studied the loading and unloading scheduling problem on a transshipment platform. They stated that timing is a crucial issue and that well-implemented platforms needed to be well scheduled. In their model, the platform was broken down into an import and an export area. They assumed that semi-trailers (contain items) entered the platform at various times. The items were broken down and either shipped directly to the export area or sent to temporary storage. At the outbound door, semi-trailers shipped according to a scheduled departure time. The objective of the problem was to schedule the unloading process so that each loading semi-trailer was completed on its due date. They classified the problem as a two-phase parallel machine scheduling problem with a time window constraint (NP-hard). They proposed two heuristic methods as resolution approaches: a squeaky wheel embedded in a genetic algorithm and linear programming with a genetic algorithm. They concluded that both methods obtained desirable solutions.

Mc Williams et al. [14] studied the operational activities in a freight consolidation terminal. The terminal consisted of motor carriers and inbound and outbound doors. They stated that a reduction in transfer time could decrease congestion and transporter cost. Time span was defined as the interval between the first pallet unloaded and the last one loaded. The goal was to obtain the unloading schedule that minimizes the time span. Simulation-based scheduling using a genetic algorithm was proposed to solve the case.

Yu W. et al. [15] sequenced inbound and outbound semi-trailers to appropriate docks. They stated that minimizing the operation time maximizes the platform throughput. In this study, the operation time was the interval between when the first product comes off the first unloading semi-trailer and when the last product is loaded onto the last semi-trailer. Product assignments were decided along with the docking sequence of inbound and outbound semi-trailers. In this situation, an improper sequence may lead to an increase in operation time. In the model, each semi-trailer must stay at the dock until it is completely loaded or unloaded. They proposed mathematical formulation, complete enumeration and a heuristic method as a resolution approaches. Neither mathematical formulation nor complete enumeration was suitable for medium and large problems. The main idea of the heuristic algorithm was to minimize the total number of products that passed through temporary storage.

As can be seen in Table 2–2, none of the studies directly considered inventory level as a main factor. They dealt mainly with time scheduling and its influence on product flow.

In this study, we develop a model for sequencing semi-trailers and a loading and unloading policy by considering product flow and inventory level. In our model, we penalize the products that are sent to storage. Direct transferring leads to a more efficient use of transporters and a reduction in inventory level. Our goal is to increase the number of products that move directly from the inbound to the outbound door.

Table 2-2 : Research at platform level

Platform level						
Work	**[7]**	**[8][9]**	**[11][12]**	**[13]**	**[14]**	**[15]**
Objective	Compare one and two-stage systems	Labor cost reduction	Door assignment	Loading and unloading scheduling problem	Transfer time reduction	Minimizing operation time
Considered factor						
Timing	☐	☒	☒	☒	☒	☒
Product flow	☒	☒	☒	☐	☒	☒
Inventory	☐	☐	☐	☐	☐	☐
Transporters	☐	☒	☐	☐	☒	☐

Baptiste et al. [16] investigated the semi-trailer sequencing problem for platforms with a single inbound and a single outbound door. They considered wait time, products movement and semi-trailer movement (temporary movement to the parking lot and then back to the dock). Four axes were observed for the problem:

- Incoming sequence
- Outgoing sequence
- Number of semi-trailers for each destination
- The loading/unloading policy (First in first out [FIFO], Last in last out [LIFO])

Based on these axes, four cases were developed:

Case I: LIFO unloading policy

Case II: LIFO unloading policy without shifting

Case II: No restriction on unloading sequence; but semi-trailer shifting is not allowed

Case IV: General case (unloading a sequence of semi-trailers)

For each case they presented a polynomial solvable algorithm.

CHAPTER 3: MODEL DESCRIPTION AND MATHEMATICAL FORMULATIONS

In this chapter the model is described in more detail. In section 3.1 we illustrate and define the objective of the problem and assumptions. In section 3.2 we propose the mathematical formulation for the presented model.

3.1 Model description

In the proposed model, k incoming semi-trailers arrive at the inbound door and unload products going to various destinations. If the outgoing semi-trailer departs to the products' final destination, the products move directly to the outbound semi-trailer; otherwise, they transfer to temporary storage. The model is illustrated in Figure 1-1.

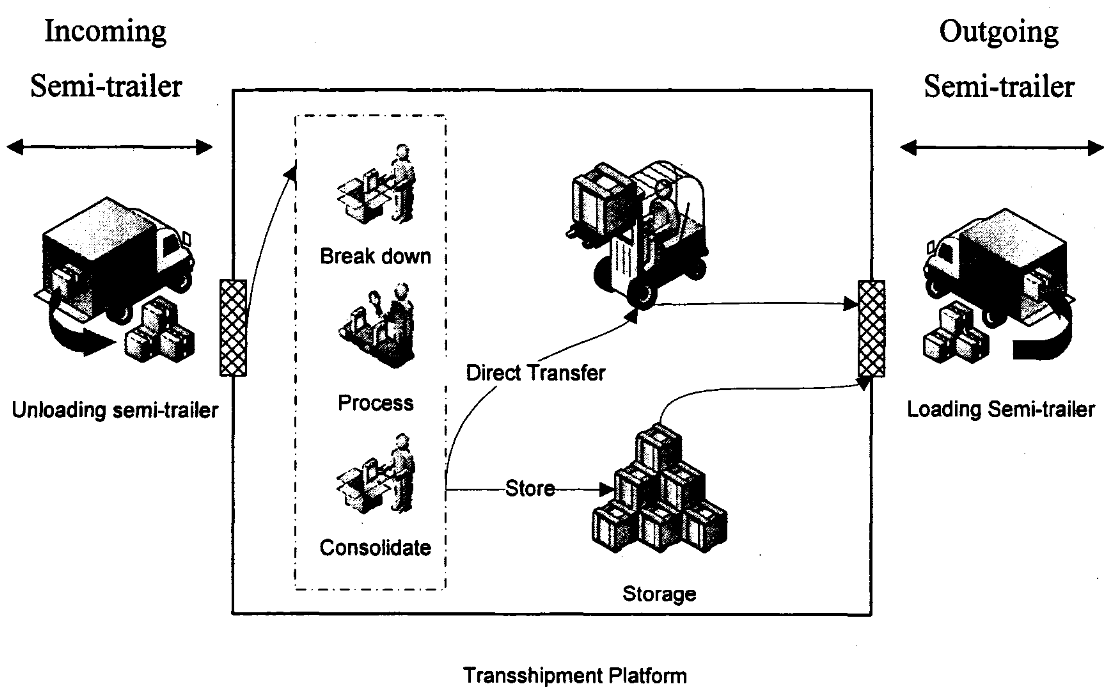

Figure 1-1: Model description

In practice, transshipment has various layouts. In this study, the layout is restricted to one receiving and one shipping door. This restriction is not realistic, but it can be used as a baseline for other layouts.

In this model, the following assumptions are considered:

- Each semi-trailer leaves the inbound door when it is fully unloaded. On the other side, each semi-trailer leaves the outbound door when it is fully loaded.
- The internal operational components of cross-docking, such as sorting and merging, are not considered.
- The storage capacity is unlimited.
- Each outbound semi-trailer departs for only one destination.
- All incoming and outgoing semi-trailers are available at time zero.
- The products differ by their destination.
- Loading, unloading and transfer time are constant and are not considered.
- The time considered can be a shift or a day.
- The numbers and the capacities of incoming and outgoing semi-trailers are equal.
- There is no rule for unloading products from semi-trailer.

3.2 Mathematical formulation

Binary variables are used to formulate the model. In this formulation the sequences of incoming and outgoing semi-trailers and the loading and unloading policy are obtained in a manner to maximize the number of directly transiting products. In this formulation, we do not have any knowledge about semi-trailer sequences. However, the incoming semi-trailers are numbered so that they can be distinguished.

Notations:

i: Products incoming order (1 to k×c)

j: Destination number

c: Semi-trailer capacity

k: Number of incoming or outgoing semi-trailers

o: Sequence of incoming semi-trailer (Max o=k)

Variables:

$$X_{i,j}\begin{cases}1 & \textit{If the product in order i labled for destination j}\\ 0 & \textit{Otherwise}\end{cases}$$

$$V_{i,j}\begin{cases}1 & \begin{array}{l}\textit{If the outgoing semi-trailer for destination j wait at outbound door,}\\ \textit{in product order i}\end{array}\\ 0 & \textit{Otherwise}\end{cases}$$

$$K_{i,j}\begin{cases}1 & \begin{array}{l}\textit{If the product in order i move directly to the current loaded semi-trailer}\\ \textit{for destination j}\end{array}\\ 0 & \textit{Otherwise}\end{cases}$$

$$Z_{i,j}\begin{cases}1 & \textit{If the loaded semi-trailer leave the platform for destination j, in product order}\\ 0 & \textit{Otherwise}\end{cases}$$

$$Y_{o,k}\begin{cases}1 & \textit{If the inbound semi-trailer number k is unloaded in order o}\\ 0 & \textit{Otherwise}\end{cases}$$

Constant:

$a_{j,k}$: The number of products for destination J in trailer K

n: total number of incoming products

Objective Function:

$$MAX \quad \sum_i \sum_j K_{ij}$$

Constraints:

$$V_{i-1,j} - V_{i,j} \le Z_{i-1,j} \qquad \forall i, \forall j \tag{1}$$

$$\sum_j V_{ij} = 1 \qquad \forall i \tag{2}$$

$$\sum_{i=(k-1)c+1}^{kc} X_{i,j} - \sum_{l=1}^{k} a_{j,l} Y_{t,l} = 0 \quad \forall j, \forall t, \forall k \qquad k \in \{1,2,...,K\}, t \in \{1,2,..,K\} \tag{3}$$

$$\sum_{l=1}^{k} Y_{k,l} = 1 \quad \forall k \tag{4}$$

$$\sum_{l=1}^{k} Y_{l,k} = 1 \quad \forall k \tag{5}$$

$$\begin{cases} K_{i,j} \le X_{i,j} \\ K_{i,j} \le V_{i,j} \end{cases} \quad \forall i, \forall j \tag{6}$$

$$\sum_j X_{ij} = 1 \qquad \forall i \tag{7}$$

$$c \sum_{p=1}^{i} Z_{p,j} - \sum_{p=1}^{i} X_{p,j} \le 0 \qquad \forall i \tag{8}$$

$$\sum_{p=i}^{q} K_{p,j} - c \sum_{p=i}^{q} Z_{p,j} \le c - 1 \qquad \forall i, \forall j, \forall q \qquad q \in \{i+1,....,n\} \tag{9}$$

The objective of the model is to maximize directly transiting products. There are three types of constraints: loading sequence, unloading sequence and control. The first two constraints determine the loading sequence of semi-trailers. Constraint (1) ensures that when the new semi-trailer starts loading (in product order i), the previous semi-trailer leaves the platform ($Z_{i-1,j}=1$). In (2), for each incoming products order, there should be one loading semi-trailer. Constraints (3) to (5) determine the incoming sequence of the semi-trailers. Constraint (3) obtains the incoming sequence of semi-trailers. Constraints (4) and (5) are complementary to (3). Constraints (6) to (9) are control constraints: constraint (6) states that when there is a directly transiting product in product order i for destination j, there exists a product in order i which transfers to destination j and there is a semi-trailer in order i which departs to destination j; (7) states that in each incoming order there is one product; (8) ensures that each semi-trailer leaves the platform when it is fully loaded; and (9) is a control constraint for variable K. It states that the number of directly transiting products does not exceed the semi-trailer capacity.

For the n incoming products, m destination and k semi-trailers, there are $4nm + k^2$ variables and $4nm + 2n + 2k + (k-1)m + \frac{n(n-1)}{2}$ constraints. The small sample problem is solved in reasonable time, but for medium to large samples, the formulation is not practical. In the next section, we propose some heuristic algorithms to solve these larger problems. We have divided our model into three cases that differ by the knowledge of incoming and outgoing sequences.

CHAPTER 4: RESOLUTION APPROACHES

In this chapter, we explain the details of the resolution approach before that we investigate the decision variables in more detail.

In the previous model, the optimal solution is affected by the following decision variables:

1. Semi-trailer incoming sequence (variable Y)

2. Semi-trailer outgoing sequence (variable V)

3. Product unloading sequence (variable X)

4. Unloading policy (variable K)

The other variable (Z) in the model is a control variable for (V), and we did not consider it as a decision variable.

The first and second variables are semi-trailers orders. For the third variable (x) , each incoming semi-trailer contains items to be shipped to different destinations. Items that can be shipped to the current outgoing semi-trailer must be unloaded first. The third variable indicates product unloading order. This variable could be a constant, due to technical constraints for the unloading operations. For example, if the products are unloaded according to the FIFO (first in first out) or LIFO (last in first out) method, the third variable is constant; otherwise, it is inconstant. For the fourth variable, suppose the following situation: an outgoing semi-trailer is positioned at the outbound door and items are waiting to be shipped to the destination. The manager can choose to reship those items or to wait until an incoming semi-trailer arrives with items that can be shipped directly to the destination. There are two extreme possible policies: either the storage products are systematically used to complete semi-trailers (less inventory), or storage items are shipped in the last semi-trailer going to their destination. The optimal policy is a combination of these two extremes.

Example:

Suppose that there are ten incoming semi-trailers with the sequence of I|II|III|IV|V|VI|VII|VIII|IX|X, and ten semi-trailers which depart to destination A, B and C with the sequence of A|B|C|A|B|C|A|B|C|A. The semi-trailer capacity is considered ten units of products. Figure 4-1 presents the assignment policy for the two extreme policies and the optimal one. The first proposed policy is to wait systematically by outgoing semi-trailer for incoming one to fill. In this policy the products which are located in storage are loaded in the last outgoing semi-trailer for each destination.

For the second policy filling the semi-trailer in shorter time has a priority. In this policy, each semi-trailer is filled by products in storage or products from unloading semi-trailer. As can be seen in figure 4-1, for the first policy the semi-trailers number IV to VII should be unloaded in order to be able to transfer 10 products directly from incoming to outgoing door, while; for second policy we transfer 3 products of semi-trailer IV directly to outgoing one and fill the rest of it by the products from storage.

Destination	A	B	C	A	B	C	A	B	C
Incoming									
I	4	3	3	4	3	3	4	3	3
II	1	7	2	1	7	2	1	7	2
III	4	3	3	4	3	3	4	3	3
IV	4	3	3	4	3	3	4	3	3
V	2	4	4	2	4	4		4	4
VI		2	4	4	2	4		2	4
VII				4	3	3			
VIII		1		7	1	2			2
IX			3	6	1	3	6		3
X	4					3	4		
Loading strategy	Optimal policy			First policy			Second policy		

Figure 4-1: loading/unloading policy

For the first policy 37 products are transferred directly from inbound to outbound door, whereas, for the second policy 51 products are transferred directly. None of these two extreme policies are optimal. In fact the optimal policy is a mixture of these two extreme policies. For example, in the optimal policy it is better to unload the semi-trailer V for destination B and send seven products directly to the outgoing semi-trailer. For the optimal policy the number of direct transiting products increases to 62.

Based on the observed variables three cases are proposed. The definition of each case is presented as follows:

Case 1: The sequences of incoming and outgoing semi-trailers are known, and variables 3 and 4 are examined.

Case 2: The sequence of incoming semi-trailers is known, and variables 2, 3 and 4 are inspected.

Case 3: No information about semi-trailer sequences is known, and variables 1, 2, 3 and 4 are examined.

4.1 First case resolution approaches

The objective in the first case is to obtain an optimal loading and unloading pattern when the sequences of incoming and outgoing semi-trailers are known. To do this, we use a graph to demonstrate all possible assignments. The graph nodes and arcs show assignment states and forthcoming possibilities, respectively. The following algorithm, which is based on dynamic programming, is used to construct the graph:

Loading/unloading algorithm (LUA):

Indices:

z: Total incoming semi-trailers

j: Loading destination

l: Total outgoing semi-trailers

N: Node Number

AD: Current loading destination

d: Destinations

Node elements:

$S^N = (H_d, Gd, C, k)$

S^N : Node number N

H_d : Vector of variables indicates possible directly transiting products for each destination

G_d: Vector of variables indicates the number of products in temporary storage for each destination

C: Gain (the total number of direct transiting products from the beginning to the current node)

k: Unloading semi-trailer order number

P_d^k : Number of products for destination d in unloading semi-trailer k

TC: Semi-trailer capacity

Start

Initialize: $H_d=P_d^1, G_d=0, c=0, k=1\ (\forall d)$ (1)

Create Node: $S^0=(H_d, G_d, c, k)$ (2)

For $j=0$ to l (3)

  For all generated node in loading destination j (4)

    Do until $H_{AD} >= TC+1$ (5)

      If $H_{AD}>=TC$ (6)

        $H_{AD}=H_{AD}-TC,\ c=c+TC,\ N=N+1$ (7)

        Create Node $S^N=(H_d, G_d, c, k)$ (8)

        Restore H_d and c to the values before if statement

      Else If $H_{AD}+G_{AD}>= TC$ (9)

        $H_{AD}=0, G_{AD}=G_{AD}-(TC-H_{AD}),$ (10)

        $c=c+H_{AD},\ N=N+1$

        Create Node $S^N=(H_d, G_d, c, k)$ (11)

        Restore H_d, G_d and c to the values before else statement

      If $k<=z$ (12)

        $H_{AD}=H_{AD} + P_{AD}^K,\ G_d=G_d+ H_d \quad (\forall d \neq AD), k=k+1$ (13)

      Else Exit the loop (14)

Call domination function (15)

END

This algorithm starts with an initial node, (1) and (2). Then, it considers all the generated nodes for the previously loading semi-trailer, (3) and (4), and for each node, it generates all forthcoming nodes for the currently loading semi-trailer: (5) to (14). In (6) to (8) the algorithm considers that the loading products are all directly transferred. In (9) to (11) the algorithm considers storage products in addition to direct products for loading. In (12) to (14) the algorithm unloads the next incoming semi-trailer.

At the end, the node in the last loading semi-trailer that has the highest cost is chosen as the final node. The path from the initial node to the selected final node is the optimal policy for loading and unloading.

In practice, two domination rules (15) are used to omit unnecessary nodes.

Domination Function:

$S^p=(H_d(p), G_d(p), c(p), k(p))$ dominate $S^q=(H_d(q), G_d(q), c(q), k(q))$ if:

1) $K(p) < K(q), H_d(p) = H_d(q)\ \forall d, and\ C(p) \geq C(q)$

2) $K(p) = K(q), C(p) = C(q)\ and \sum_d H_d(q) \leq \sum_d H_d(p)$

The first rule states that, if for each destination, directly transiting products for two nodes are equal, the node with the highest cost dominates the other. The second rule applies when two or more nodes have the same cost, in which case the node with higher summation of directly transiting products for all destinations dominates the others.

Example:

Suppose that there are five incoming and five outgoing semi-trailers with a capacity of ten products each. The outgoing semi-trailers depart to three destinations (2 to destination A, 2 to destination B and 1 to destination C). The outgoing sequence is A-B-A-B-C, and the incoming sequence is I-II-III-IV-V. Table 4-1 presents the products in each trailer; for example, the first unloading trailer has six products for destination A, three for B and one for C.

Table 4-1: Example

Incoming semi-trailer	Semi-trailer contents for each destination		
	A(products)	B(products)	C(products)
I	6	3	1
II	2	7	1
III	5	2	3
IV	3	4	3
V	4	4	2

The loading/unloading algorithm is applied and the generated graph is presented in Figure 4-2. Node 1 is the initial node. The algorithm reads the initial node and generates node 2 with the cost of 10 as a possible assignment for the first loading semi-trailer. From node 2, nodes 3, 4 and 5 are generated for the second loading semi-trailer. This procedure continues until all outgoing semi-trailers are loaded. Nodes 3 and 5 have the same number of directly transiting products and the cost of node 3 is higher than node 5, but it did not satisfy the domination rule; on the other side, node 10 is dominated by node 9. In the fifth loading semi-trailer, node 12 has the highest cost (29), therefore, it is chosen as the final node and the path with nodes 1-2-3-6-9-12 is the optimal loading/unloading policy.

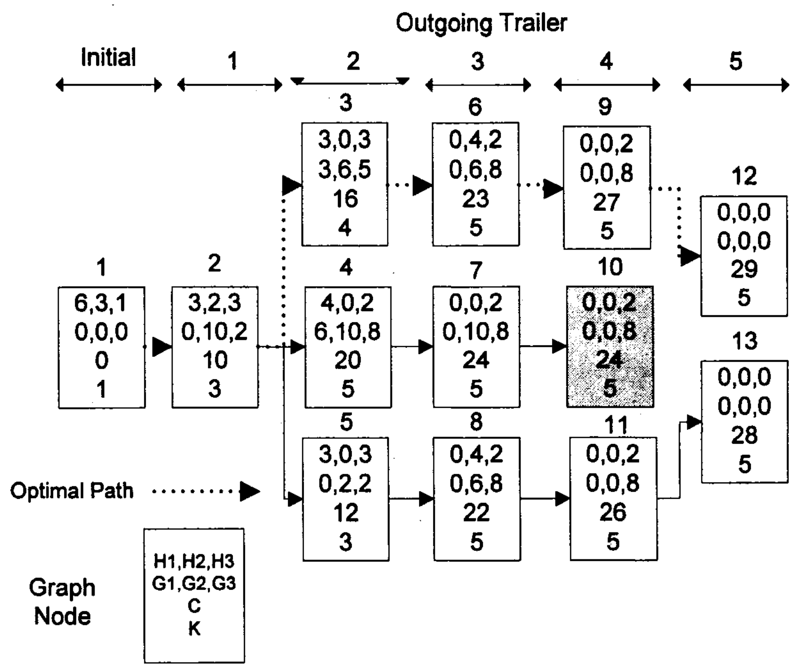

Figure 4-2: Optimal policy algorithm for the example

4.2 Second case resolution approaches

In this case, two methods are proposed: Tabu search integrated with the loading/unloading algorithm and a heuristic method. The first method is proposed by the following algorithm.

<u>**Sequencing outgoing algorithm (SOA):**</u>

Step 1: Run loading/unloading algorithm for the initial value for the number of direct transiting products

Step2: Select two loading semi-trailers' order numbers

Step3: Swap the order numbers and save them in Tabu list (if they are not for the same destination and are not already in Tabu list)

Step 4: Run loading/unloading algorithm

Step 5: Save the cost and, if it is improved, save the optimal path

Step 6: Go to step 2 or stop if the cost is not modified after 20 iterations

In the LUA and SOA, the optimal solution is not the combination of best solutions. In the other words, sometimes selecting the node with the lower cost would lead to the node with the highest cost. For example, node 4 has a higher cost than node 3 However, selecting node 3 leads to the node with the highest cost. In the proposed heuristic method, it is supposed that sub-optimal assignments lead the process to the optimal result.

The proposed algorithm is based on the greedy method, and it assumes that the sequence of incoming semi-trailers is known a priori. In each iteration, the destination that has the highest cost is selected as the outgoing destination. At the end, the selected destinations are the outgoing semi-trailer sequence.

Sequencing outgoing greedy algorithm (SOGA):

Step 1: Create an initial value (the value of directly transiting products for each destination is equal to the first unloaded semi-trailer's products; the rest are zero)

Step 2: For all loading semi-trailers do steps 3 to 8

Step 3: For all destinations do steps 4 to 7

Step 4: Do this as long as total directly transiting product for the selected destination is more than semi-trailer capacity

Step 5: Calculate the cost for current selected destination; preserve the results if improved

Step 6: Consider the next unloaded semi-trailer and update values; go to step 4

Step 7: Save best assignment

Step 8: Set best obtained outgoing assignment as current assignment; go to step 2

Step 9: The final list is the optimal outgoing assignment

Example:

In the previous example, for the given incoming sequence, the solution obtained with the SOA is 34 with the sequence B-A-B-C-A. For the SOGA, the obtained sequence is B-A-C-B-A with the cost of 33.

4.3 Third Case resolution approaches

The previous methods were developed to obtain the optimal sequence of loading and unloading semi-trailers. In the current method, Tabu search is integrated with the SOA or the SOGA to obtain the best loading sequence.

Sequencing incoming/outgoing algorithm (SIOA)/ Sequencing incoming/outgoing greedy algorithm (SIOGA):

Step 1: Run SOA (or SOGA) for initial value for the number of direct transiting product

Step2: Select two unloading semi-trailers' order numbers

Step3: If they are not in Tabu list, swap the order numbers and save them in Tabu list

Step 4: Run SOA or SOGA

Step 5: Save the cost and, if it increases, save the optimal path

Step 6: Go to step 2 or stop if the cost is not improved after 20 iterations

Example:

For the example, the cost obtained with the first algorithm is 38 with outgoing sequence B-A-B-C-A and incoming sequence II-I-V-IV-III. For the second algorithm (SIOGA), the outgoing sequence is A-B-A-C-B with I-V-II-III-IV as an incoming sequence and a cost of 37.

CHAPTER 5: EXPERIMENTS

In this chapter we have done some experiments to test the proposed algorithms. For the experiments we consider two criteria for defined cases : number of semi-trailers and number of destinations. Therefore, we define small, medium and large size problems and for each problem size we introduce two or three numbers of destinations. Then we generate 20 data samples for each defined case. In section 5.1 the algorithm performances are discussed and in section 5.2 we investigate the algorithm running time.

5.1 Data

In the previous section, the resolution approaches for semi-trailer sequencing problems in transshipment platform were studied. We discussed that sequencing semi-trailers is one of the platform operational activities that affect performance. In this section we test the proposed algorithms. We consider small (5 semi-trailers), medium (10 semi-trailers) and large (20 semi-trailers) problems. For small problems, three and five destinations are defined. For medium and large problems, three, five and ten destinations are defined. We assume that the outgoing semi-trailers are equally distributed between destinations. For example, for the medium-sized problem with three destinations, we had four semi-trailers depart to destination I and three semi-trailers depart to destinations II and III, respectively. Table 5-1 presents the distribution of trailers between destinations.

The capacities of incoming and outgoing semi-trailers are 10 units of products. For each defined problem, 20 sets of data are randomly generated.

Table 5-1: Distribution of semi-trailers for each case

Problem size	Number of destinations	Semi-trailer per destination I	II	III	IV	V	VI	VII	VII I	IX	X
Small	3	2	2	1							
	5	1	1	1	1	1					
Medium	3	4	3	3							
	5	2	2	2	2	2					
	10	1	1	1	1	1	1	1	1	1	1
Large	3	7	7	6							
	5	4	4	4	4						
	10	2	2	2	2	2	2	2	2	2	2

The resolution approaches for the first, second and third cases are tested with generated samples. We completely enumerated all the possible incoming and outgoing sequences for the small problem to obtain the best and the worst bound; we also calculated the average and the standard deviation of each defined problem. Tables 5-2, 5-3 and 5-4 present the results for the small problem.

Table 5-2: The best and the worst bound

	Second case				Third case			
Destination combination	Average	Std	WB*	BB**	Average	Std	WB*	BB**
2-2-1	25.96	4.85	18.2	33.45	25.81	4.93	16.15	37.9
1-1-1-1-1	18.19	1.35	16.6	20.85	18.14	1.31	15.75	23.85

* WB: the average obtained worse bound
** BB: the average of obtained best bound

Table 5-3: Average Number of direct transiting products (Small problem)

(Max 50 product)

Destination combination	L.U.A	S.O.A	S.O.G.A	S.I.O.A	S.I.O.G.A*
2-2-1	30.4	33.3	32.8	37.4	37
1-1-1-1-1	18.3	20.65	20.55	23.45	23.3

* L.U.A: Loading/unloading algorithm
S.O.A: Sequencing outgoing algorithm
S.O.G.A: Sequencing outgoing greedy algorithm
S.I.O.A: Sequencing incoming/outgoing algorithm
S.I.O.G.A: Sequencing incoming/outgoing greedy algorithm

Table 5-4: Algorithm performance

Second case

Destination combination	WB&BB	S.O.A & L.U.A	S.O.A & BB	S.O.G.A & L.U.A	S.O.G.A & BB
(2-2-1)	30.50%	5.80%	0.30%	4.80%	1.30%
(1-1-1-1-1)	8.50%	4.70%	0.40%	4.50%	0.60%

Third case

	WB&BB	S.I.O.A & L.U.A	S.I.O.A & BB	S.I.O.G.A & L.U.A	S.I.O.G.A & BB
(2-2-1)	43.50%	14.00%	1.00%	13.20%	1.80%
(1-1-1-1-1)	16.20%	10.30%	0.80%	10.00%	1.10%

Table 5-2 shows the best and the worst bound and the average number of directly transiting products. The average number of directly transiting products does not change notably. The small standard deviations indicate that although the average number of directly transiting products in the second case did not differ from the third case, the best and the worst bounds change. The gap between them is increased from 30.50% to 43.50% for 2-2-1 and 8.50% to 16.20% for 1-1-1-1-1 (Table 5-4).

Table 5-3 presents the average number of directly transiting products for each case. The solution for the LUA is an initial answer, which is used as a base to compare the improvements for each case. The improvements of each algorithm (percentages) are presented in Table 5-4. As can be seen, the improvements of applying greedy algorithms are less than the algorithms based on the LUA, but their gaps with the optimal solution are negligible (0.40% compare to 0.60% for case 1-1-1-1-1). As shown in Table 5-4, the gap between the optimal solution and the algorithms solution is less than 2%, which indicate that in our sample problem both algorithms have good performance. Figures 5-1 and 5-2 illustrate the solutions for each data set in case 2-2-1 of the small problem.

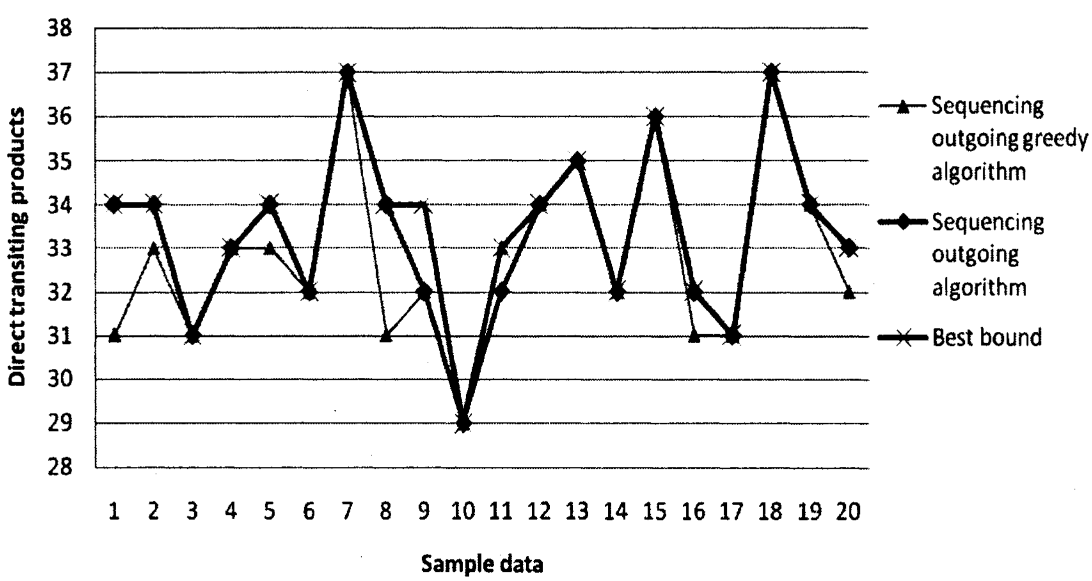

Figure 5-1: Comparing the algorithms' performance for second case with optimal solution (case 2-2-1 data)

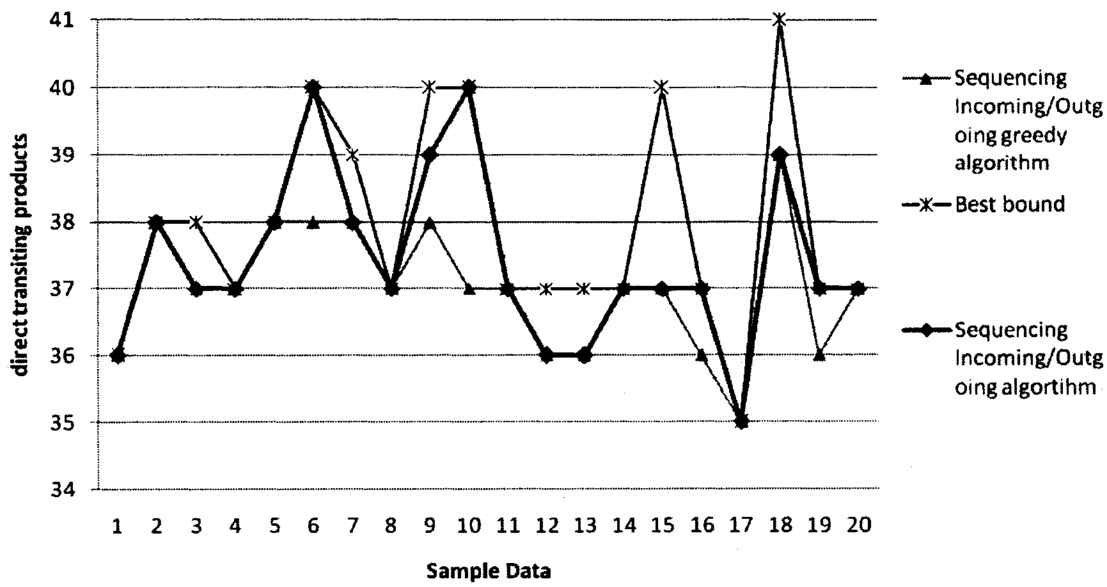

Figure 5-2: Comparing algorithms' performances for third case with optimal solution (case 2-2-1 data)

From Table 5-4, we see that sequencing both incoming and outgoing semi-trailers notably increased the performance. For example the improvement in case 2-2-1 is 8.2% (from 5.80% to 14.00%) for LUA-based algorithms and 8.4% (from 4.80% to 13.20%) for greedy-based algorithms.

The results of the algorithm for medium and large problems are in Table 5-5. Sequencing both incoming and outgoing semi-trailers notably increases the number of directly transiting products; in fact, it synchronizes the product flow from the receiving to the shipping door. Figures 5-3, 5-4, 5-5 illustrate the improvements in the small, medium and large problems. In all cases, sequencing both incoming and outgoing semi-trailers notably increases performance.

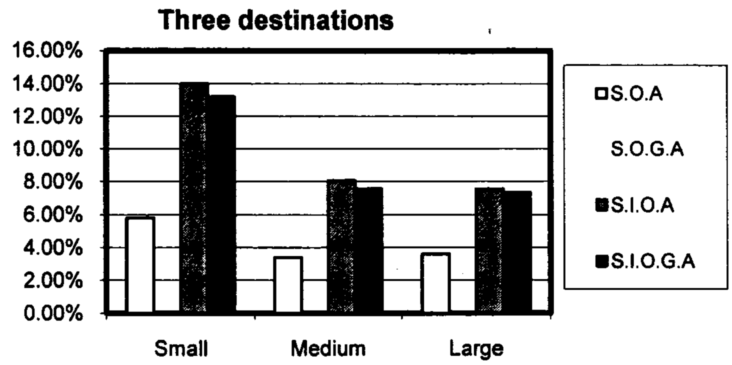

Figure 5-3: Compare algorithms performance for three destinations

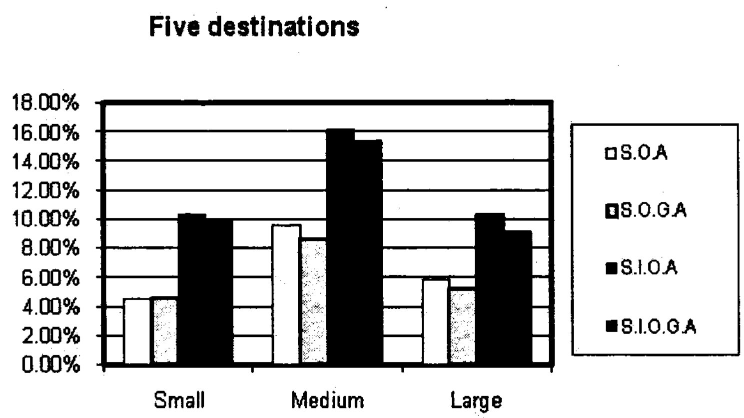

Figure 5-4: Compare algorithms performance for five destinations

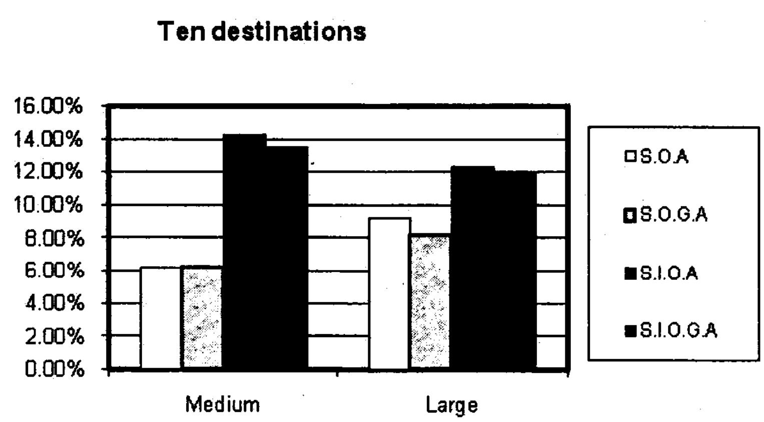

Figure 5-5 : Compare algorithms performance for ten destinations

Table 5-5: Algorithms improvements for five,ten and twenty semi-trailers

Problem size	Algorithm	3	5	10
		Number of destinations		
Small	S.O.A	5.80%	4.50%	N/A
	S.O.G.A	4.80%	4.50%	N/A
	S.I.O.A	14.00%	10.30%	N/A
	S.I.O.G.A	13.20%	10.00%	N/A
Medium	S.O.A	3.40%	9.60%	6.25%
	S.O.G.A	1.80%	8.65%	6.20%
	S.I.O.A	8.05%	16.10%	14.30%
	S.I.O.G.A	7.55%	15.35%	13.50%
Large	S.O.A	3.60%	5.83%	9.25%
	S.O.G.A	2.60%	5.15%	8.15%
	S.I.O.A	7.53%	10.30%	12.28%
	S.I.O.G.A	7.33%	9.10%	12.05%

To conclude, for our generated samples, all of the algorithms noticeably increased performance, but the greedy-based algorithms had fewer improvements than the others. Also, for our small problem, the gap between the optimal and the algorithms solution was less than 2%.

5.2 Model

In this section, the loading and unloading algorithm is investigated in more detail. We consider the number of generated nodes as the criterion for the algorithm's running time, and we examine the relationship between problem size and number of destinations. We selected a sample problem and varied one variable at the time. The sample has ten incoming and ten outgoing semi-trailers (each with the capacity of ten products) and three destinations.

<u>The relationship between number of products and number of generated nodes</u>

We studied the relationship between problem size and running time. The size of the problem depends on the number of incoming products; therefore, we consider the number of products as a variable. The sample problem was run with different numbers of products, and the results are shown in Table 5-6 and Figure 5-6.

Table 5-6: Number of generated nodes for different numbers of products

Number of products	50	100	200	400	800
Generated nodes	10	120	1107	6888	31606

The number of generated nodes increases exponentially with an increase in the number of products. In fact, the algorithm could be classified as a "shortest path problem" with resource constraints, as its complexity is exponentially related to the number of products.

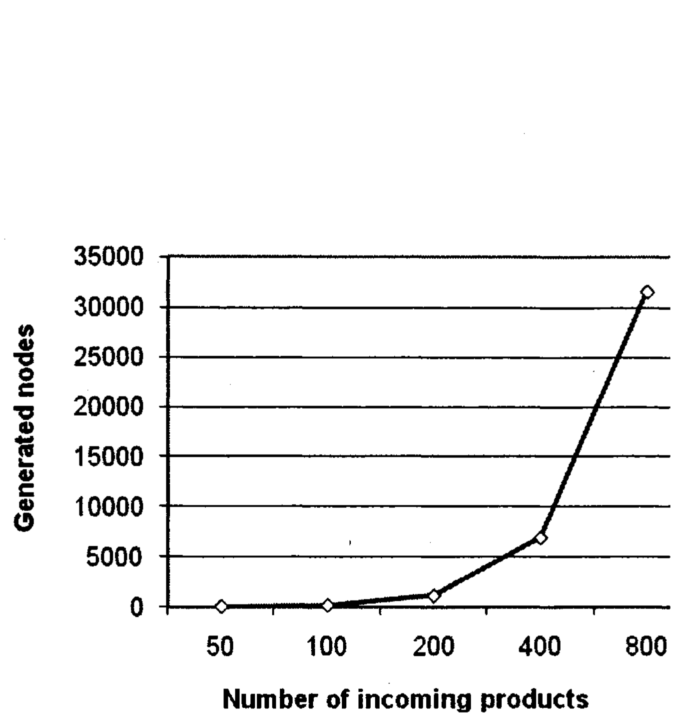

Figure 5-6: Number of generated nodes for different numbers of products

<u>The relationship between number of destinations and number of generated nodes</u>

To examine the relationship between the number of destinations and the number of generated nodes, we varied the number of destinations in our sample between 2 and 10. We considered the semi-trailers to be equally distributed between destinations. For better results, we considered 30 random incoming sequences. The results are shown in Table 5-7 and Figures 5-7 and 5-8.

Table 5-7: Generated nodes with different numbers of destinations

Number of destinations	2	3	4	5	6	7	8	9	10
Average	81.4	99.06	61.86	65.14	39.53	28.28	28.90	11.41	11.87
standard deviation	19.39	29.12	23.93	23.48	17.72	12.18	8.70	1.36	0.99

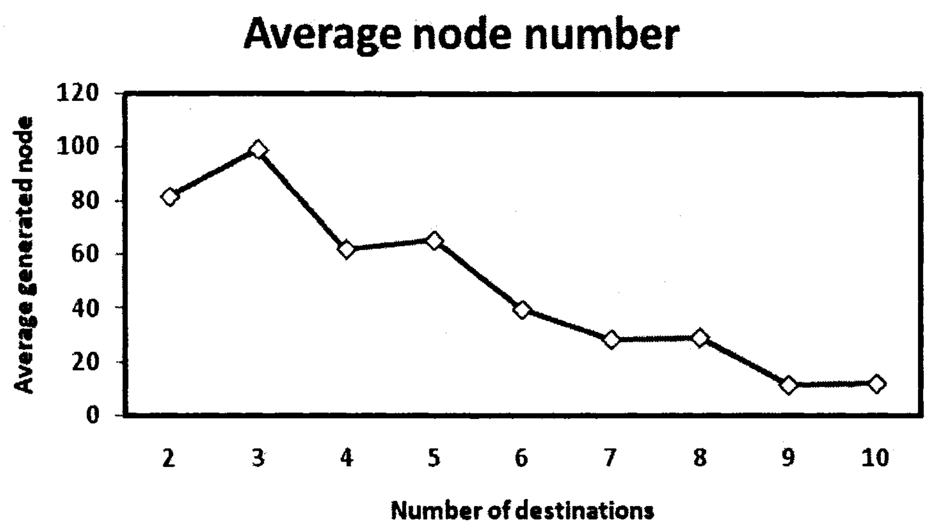

Figure 5-7: Average number of generated nodes for different destinations

Figure 5-8 shows that the number of generated nodes generally decreases when the number of destinations increases. The same trend is seen for standard deviation, where the fluctuation of generated nodes is decreased.

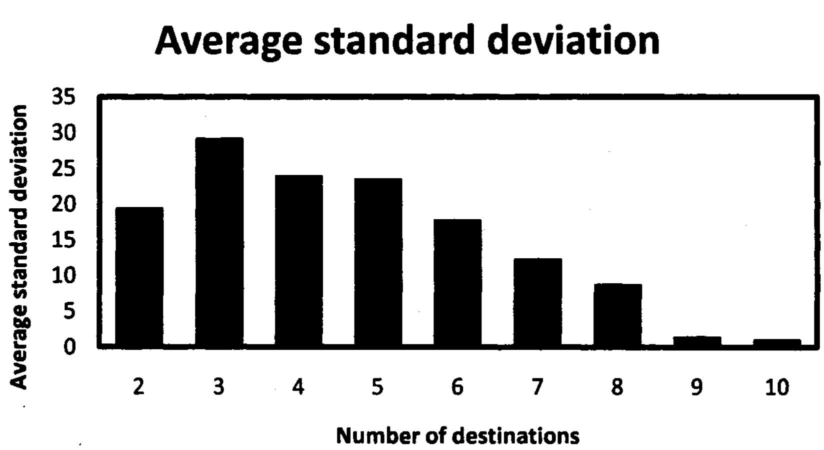

Figure 5-8: The standard deviation of the average generated nodes for different destinations

CHAPTER 6: CONCLUSION AND DISCUSSION

The transshipment platform is where products from incoming semi-trailers are unloaded and then consolidated with other products for reshipment. The efficiency of such platforms is manipulated by the ratio of directly transiting products to total transferring products.

This research explores the particular case of a platform with a single receiving and a single shipping door. Based on our knowledge about the sequences, three cases of the problem were studied. Dynamic programming (LUA) and a heuristic method (SOGA) were proposed as two major functions to solve the cases. Experiments were performed on small, medium and large problems. In addition, we did complete enumeration for the small problem to investigate the performance of the proposed algorithms.

When both sequences were known, an algorithm based on dynamic programming found the optimal use of the temporary inventory. When incoming sequences were known, two different approaches were proposed. The first used a dynamic programming for optimal loading/unloading and evaluation functions to determine outgoing sequence. For the second, we proposed a fast greedy-based heuristic algorithm. In our generated problems, the gap between the two proposed algorithms is negligible. For the third case, the two previously proposed algorithms were developed with evolutionary search for incoming sequence. Both algorithms present acceptable performance and show notable improvement compared to the second case.

Overall, sequencing incoming and outgoing semi-trailers is one of the operational activities on a cross-docking platform that can increase transshipment performance by increasing the number of directly transiting products. In this work, we assume one inbound and one outbound door. This model should be extended to a general platform with multiple incoming and outgoing doors.

REFFERENCES

[1] KINNEAR. E. (1997), Is there ant Magic in cross-docking. *Supply chain management: an international Journal*, 2(2):49-52.

[2] ZHANG. M. (1997), Cross-docking network design, PhD thesis in Industrial and system engineering , Georgia institute of technology.

[3] RATLIFF HD., VATE JV, ZHANG M. Network design for load-driven cross-docking systems, *http://www.isye.gatech.edu/apps/research-papers/papers/misc9914.pdf*

[4] DONALDSON H., JOHNSON EL., RATLIFF HD., ZHANG M, Schedule-driven cross-docking networks, *http://www.isye.gatech.edu/apps/research-papers/papers/misc9904.pdf*

[5] CHEN P., GUO Y., LIM A., RODRIGUES B. (2006),Multiple cross docks with inventory and time windows, *Computers and Operations Research* 33, 43–63.

[6] LEE Y.H., JUNG J.W., LEE K.M. (2006), Vehicle routing scheduling for cross-docking in the supply chain, *Computers and Industrial Engineering*, article in press.

[7] GUE K.R., KANG K. (2001), Staging queues in material handling and transportation systems, *Proceedings of the 2001 Winter Simulation Conference*. 1104-1108.

[8] BARTHOLDI J.J, GUE KR. (2000) The best shape for a cross-dock. *INFORMS national conference*, San Antonio, TX.

[9] BARTHOLDI J. J., GUE K. R., (2004), The Best Shape for a Cross-dock, *Transportation Science*, 38 (2), 235-244.

[10] SCHAFFER B. (1997), Implementing a Successful Cross-docking Operation, *IIE Solutions*, 29 (10), 34-37.

[11] TSUI L. Y., CHANG C. H. (1992), Optimal Solution to a Dock Door Assignment Problem, *Computers and Industrial Engineering*, 23, 283-286.

[12] TSUI L. Y., CHANG C. H.(1990), A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards, *Computers and Industrial Engineering*, 19, 309-312.

[13] LI Y., RODRIGUES B.(2004), Cross-docking —JIT scheduling with time windows, *Journal of the Operational Research Society 55*, 1342–1351.

[14] MC WILLIAMS D.L., STANFIELD P.M., GEIGER C.D.(2005), The parcel hub scheduling problem: A simulation-based solution approach, *Computers and Industrial Engineering, 49*, 393–412.

[15] YU W., EGBELU P.J.(2006), Scheduling of inbound and outbound trucks in cross-docking system with temporary storage, *European journal of operation research*, Article in press.

[16] P. BAPTISTE, B. PENZ, R. LARBI (2007), Polynomial time solution methods for some cross-docking problems, *International Conference on Industrial Engineering and Systems Management.*

[17] BAPTISTE P., MAKNOON M.Y (2007), Ordonnancement simultané des flots entrant et sortant d'une plateforme simple de transbordement, *7e Congrès international de génie industriel*, Trois-Rivières, Québec.

[18] BAPTISTE P., MAKNOON M.Y., (2007), Cross-docking: scheduling of incoming and outgoing semi-trailers,*19th International Conference on Production Research*, Valparaiso, Chile

APPENDIX A:

SIMULTANEOUS SEQUENCING OF INCOMING AND OUTGOING SEMI-TRAILERS ON A CROSS-DOCKING PLATFORM
(To be send to international journal of production and economics)

M.Y. Maknoon, P. Baptiste

Département de mathématiques et génie industriel, École polytechnique
Montréal
Montréal, Québec, Canada

Abstract

This paper presents the semi-trailer scheduling problem on a cross-docking platform. There are sequences of inbound and outbound semi-trailers on the transshipment platform. At the inbound door, the products are unloaded, broken down, processed and consolidated for reshipment at the outbound door. The unloaded products transfer directly to the outbound door if they move to the currently loading semi-trailer; otherwise, the products are put into temporary storage for future reshipment. Increasing the proportion of directly transiting products leads to a reduction in inventory levels and a more efficient utilization of platform transporters. In this work, the simultaneous sequencing of inbound and outbound semi-trailers is studied to maximize the direct flow of products between receiving and shipping doors. Three cases of this problem are studied. The cases vary by the amount of available knowledge about incoming and outgoing sequences. Dynamic programming, Tabu search and heuristic methods are proposed to solve the cases. In addition, experiments are performed, which show that sequencing both incoming and outgoing semi-trailers increases platform performance.

Keywords:
cross-docking, scheduling, sequencing incoming and outgoing semi-trailers, Tabu search, heuristic methods

1. INTRODUCTION

In a supply chain, the cross-docking platform is a logistic facility between producer and consumer with the advantage of a just-in-time environment. This advantage helps enterprises using a cross-docking facility to reduce their logistic costs.

The sequence of semi-trailers plays an important role in platform performance. The cross-docking platform consists of an inbound door, temporary storage and an outbound door. At the inbound door, the incoming products, which differ by their sending destinations, are unloaded, broken down, processed and consolidated for reshipment at the outbound door. The consolidated products are either transferred directly to the loading semi-trailer or put into temporary storage for future reshipment. Transferring products directly leads to less inventory and reduced transporter utilization, both of which contribute to platform performance. One of the ways to increase performance of a platform is to sequence inbound and outbound semi-trailers in a way that maximizes the direct flow of products between doors.

This paper studies three cases of the sequencing problem in transshipment. The cases differ by the available knowledge about the incoming or outgoing sequences. Dynamic programming and heuristics are used. These general methods are integrated with Tabu search as a resolution for the cases. The results indicate that sequencing incoming and outgoing semi-trailers notably increases the ratio of directly transiting products to total transferred products, which leads to a reduction in inventory level and transporter utilizations. We conclude that sequencing semi-trailers is one of the platform operational activities that can increase cross-docking performance.

1.1. Literature review

The basic activities in traditional warehouse systems are receiving, storing and shipping of the products. Storing and picking up orders are costly, making cross-docking, which features a just-in-time environment, an attractive solution. Kinnear [1] defined cross-docking as a transshipment platform that receives products from a supplier for several destinations and consolidates them with other suppliers' products for a common final delivery destination.

Transshipment performance is generally studied at two levels. At the global level, cross-docking networks are studied. The research is mainly aimed at designing a network to reduce the number of semi-trailers or the inventory level. At the platform level, cross-docking itself is considered, with platform layout, operational management, and other activities as variables that could increase platform efficiency.

The number and location of platforms play important roles in supply chain global performance. Zhang M. [2] defined two strategies for dispatching semi-trailers: load-driven and schedule-driven. In load-driven systems, the semi-trailers dispatch when there are sufficient products on the platform. In contrast, in schedule-driven systems, the semi-trailers depart at scheduled times without considering the number of products. Ratliff et al. [3] studied a load-driven transshipping network for the automobile industry to obtain the number and location of each platform and the shipping flow between them. To reduce the inventory, they minimized the number of semi-trailers on the platform. A similar study was done on the US postal service by Donaldson et al. [4], examining a schedule-driven network. Ping Chen et al. [5] extended the above research to minimize global transportation costs by considering pickup and delivery time windows, platform storage capacity and inventory level. Lee et al. [6] reduced inventory level and delivery lead-time in a cross-docking network by integrating the transshipment model and semi-trailer scheduling. The key point in their research was to integrate the arrival and consolidation of products. In their integrated model, all products moved from supplier to customer without any interruptions.

At the global level, platform performance problems, such as how semi-trailers are assigned to doors or how well the products transfer between doors, have not been studied. At the platform level, product flow is the main variable considered to increase platform efficiency. Designing the platform layout and managing operational activities have been studied as methods to control product flow.

Gue and Kang [7] studied a simulated platform and concluded that a two-stage system (loading and unloading) has lower output than a single-stage system when the freights are blocked between stages. Bartholdi and Gue [8] minimized labor cost by developing models for travel cost within the docks and the congestion that occurs during consolidation. In [9], they state that freight flow patterns are determined by platform layout, geometry, material handling systems, freight mix and scheduling.

Schaffer [10] suggested that operational management is one of the requirements for successful cross-docking. The first work on platform operational activities was done by Tsui and Chang [11], who formulated the dock assignment problem, simultaneously allocating both inbound and outbound doors to semi-trailers, as an integer programming model. They proposed a microcomputer-based decision support tool to assign dock doors in a freight yard. [12].

Li et al. [13] studied the loading/unloading scheduling problem on a transshipment platform when each container had to be filled at an exact time. They used machine scheduling to model the problem. In their model, the transshipment platform is divided into loading and unloading areas. The arrival dates for incoming semi-trailers are variable. The received items are then either shipped away directly or sent to the exportation area to be loaded for reshipment. In their problem, the time to start unloading is scheduled so that each loading semi-trailer is completed on its due date.

Platform transfer operations in the parcel industry were studied by D.L. McWilliams et al. [14]. They considered the platform as an inbound door, an outbound door and a conveyor to transfer parcels between them. They minimized the time interval between the first unloaded parcel and the last loaded one.

Yu W. et al. [15] minimized the completion time by scheduling inbound and outbound semi-trailers when the storage is located at shipping dock. They proposed a mathematical formulation for small samples and a heuristic method for large samples.

Sequencing the semi-trailers is one of the most important operations in cross-docking. Baptiste et al. [16] classified the semi-trailer sequencing problem for transshipment platform into four classes with the following criteria:

- Knowledge of inbound and outbound sequences
- Number of semi-trailers for each destination
- Queuing model

They suggested that the four classes are polynomial solvable. This paper studies the sequencing problem with different amounts of knowledge about incoming and outgoing semi-trailers.

2. MODEL DESCRIPTION AND ASSUMPTIONS

In the proposed model, K incoming semi-trailers arrive at the inbound door and unload products going to various destinations. If the outgoing semi-trailer departs to the products' final destination, the products move directly to the outbound semi-trailer; otherwise, they transfer to temporary storage. The model is illustrated in Figure 2-1.

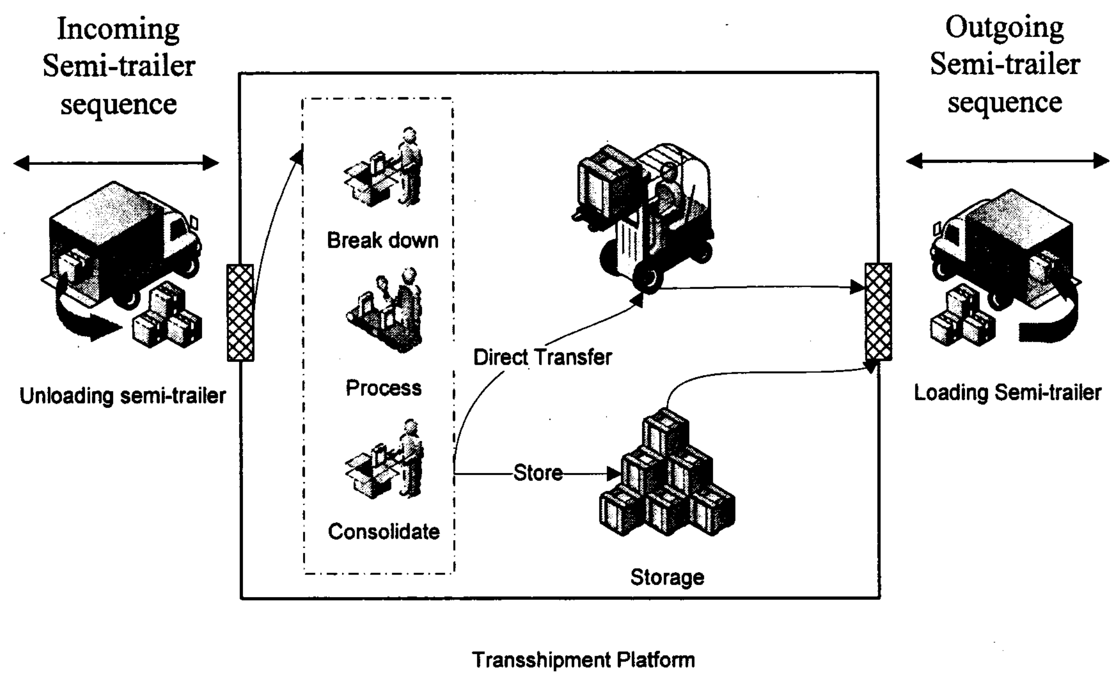

Figure 2-1: Model description

In practice, transshipment has various layouts. In this study, the layout is restricted to one receiving and one shipping door. This restriction is not realistic, but it can be used as a baseline for other layouts.

In this model, the following assumptions are considered:

- Each semi-trailer leaves the inbound door when it is fully unloaded. On the other side, each semi-trailer leaves the outbound door when it is fully loaded.
- The internal operational components of cross-docking, such as sorting and merging, are not considered.
- The storage capacity is unlimited.
- Each outbound semi-trailer departs for only one destination.
- All incoming and outgoing semi-trailers are available at time zero.
- The products differ by their destination.
- Loading, unloading and transfer time are constant and are not considered.
- The time considered can be a shift or a day.
- The numbers and the capacities of incoming and outgoing semi-trailers are equal.
- There is no rule for unloading products from semi-trailer.

2.1. Mathematical Formulation

Binary variables are used to formulate the model. In this formulation the sequences of incoming and outgoing semi-trailers and the loading and unloading policy are obtained in a manner to maximize the number of directly transiting products. In this formulation, we do not have any knowledge about semi-trailer sequences. However, the incoming semi-trailers are numbered so that they can be distinguished.

Notations:

i: Products incoming order (1 to $k \times c$)

j: Destination number

c: Semi-trailer capacity

k: Number of incoming or outgoing semi-trailers

o: Sequence of incoming semi-trailer (Max o=k)

Variables:

$$X_{i,j}\begin{cases}1 & \textit{If the product in order i labled for destination j}\\ 0 & \textit{Otherwise}\end{cases}$$

$$V_{i,j}\begin{cases}1 & \begin{array}{l}\textit{If the outgoing semi-trailer for destination j wait at outbound door,}\\ \textit{in product order i}\end{array}\\ 0 & \textit{Otherwise}\end{cases}$$

$$K_{i,j}\begin{cases}1 & \begin{array}{l}\textit{If the product in order i move directly to the current loaded semi-trailer}\\ \textit{for destination j}\end{array}\\ 0 & \textit{Otherwise}\end{cases}$$

$$Z_{i,j}\begin{cases}1 & \textit{If the loaded semi-trailer leave the platform for destination j, in product order}\\ 0 & \textit{Otherwise}\end{cases}$$

$$Y_{o,k}\begin{cases}1 & \textit{If the inbound semi-trailer number k is unloaded in order o}\\ 0 & \textit{Otherwise}\end{cases}$$

Constant:

$a_{j,k}$: The number of products for destination J in trailer K

n: total number of incoming products

Objective Function:

$$MAX \quad \sum_{i}\sum_{j} K_{ij}$$

Constraints:

$$V_{i-1,j} - V_{i,j} \leq Z_{i-1,j} \qquad \forall i, \forall j \tag{1}$$

$$\sum_j V_{ij} = 1 \qquad \forall i \tag{2}$$

$$\sum_{i=(k-1)c+1}^{kc} X_{i,j} - \sum_{l=1}^{k} a_{j,l} Y_{t,l} = 0 \quad \forall j, \forall t, \forall k \qquad k \in \{1,2,...,K\}, t \in \{1,2,..,K\} \tag{3}$$

$$\sum_{l=1}^{k} Y_{k,l} = 1 \quad \forall k \tag{4}$$

$$\sum_{l=1}^{k} Y_{l,k} = 1 \quad \forall k \tag{5}$$

$$\begin{cases} K_{i,j} \leq X_{i,j} \\ K_{i,j} \leq V_{i,j} \end{cases} \quad \forall i, \forall j \tag{6}$$

$$\sum_j X_{ij} = 1 \qquad \forall i \tag{7}$$

$$c\sum_{p=1}^{i} Z_{p,j} - \sum_{p=1}^{i} X_{p,j} \leq 0 \qquad \forall i \tag{8}$$

$$\sum_{p=i}^{q} K_{p,j} - c\sum_{p=i}^{q} Z_{p,j} \leq c-1 \qquad \forall i, \forall j, \forall q \qquad q \in \{i+1,....,n\} \tag{9}$$

The objective of the model is to maximize directly transiting products. There are three types of constraints: loading sequence, unloading sequence and control. The first two constraints determine the loading sequence of semi-trailers. Constraint (1) ensures that when the new semi-trailer starts loading (in product order i), the previous semi-trailer leaves the platform ($Z_{i-1,j}=1$). In (2), for each incoming products order, there should be one loading semi-trailer. Constraints (3) to (5) determine the incoming sequence of the semi-trailers. Constraint (3) obtains the incoming sequence of semi-trailers. Constraints (4) and (5) are complementary to (3). Constraints (6) to (9) are control constraints: constraint (6) states that when there is a directly transiting product in product order i for destination j, there exists a product in order i which transfers to destination j and there is a semi-trailer in order i which departs to destination j; (7) states that in each incoming order there is one product; (8) ensures that each semi-trailer leaves the platform when it is fully loaded; and (9) is a control constraint for variable K. It states that the number of directly transiting products does not exceed the semi-trailer capacity.

For the *n* incoming products, *m* distinct variables and *k* semi-trailers, there are $4nm + k^2$ variables and $4nm + 2n + 2k + (k-1)m + \frac{n(n-1)}{2}$ constraints. The small sample problem is solved in reasonable time, but for medium to large samples, a mathematical model is not practical. In the next section, we propose some heuristic algorithms to solve these larger problems. We have divided our model into three cases that differ by the knowledge of incoming and outgoing sequences.

3. RESOLUTIONS:

In the previous model, the optimal solution is affected by the following decision variables:

1. Semi-trailer incoming sequence (variable Y)

2. Semi-trailer outgoing sequence (variable V)

3. Product unloading sequence (variable X)

4. Unloading policy (variable K)

The other variable (Z) in the model is a control variable for (V), and we did not consider it as a decision variable.

The first and second variables are semi-trailers orders. For the third variable, each incoming semi-trailer contains items to be shipped to different destinations. Items that can be shipped to the current outgoing semi-trailer must be unloaded first. The third variable indicates product unloading order. This variable could be a constant, due to technical constraints for the unloading operations. For example, if the products are unloaded according to the FIFO (first in first out) or LIFO(last in first out) method, the third variable is constant; otherwise, it is inconstant. For the fourth variable suppose the following situation: an outgoing semi-trailer is positioned at the outbound door and items are waiting to be shipped to the destination. The manager can choose to reship those items or to wait until an incoming semi-trailer arrives with items that can be shipped directly to the destination. There are two extreme possible policies: either the storage products are systematically used to complete semi-trailers (less inventory), or storage items are shipped in the last semi-trailer going to their destination. The optimal policy is a combination of these two extremes.

Three cases are proposed:

Case 1: The sequences of incoming and outgoing semi-trailers are known, and variables 3 and 4 are examined.

Case 2: The sequence of incoming semi-trailers is known, and variables 2, 3 and 4 are inspected.

Case 3: No information about semi-trailer sequences is known, and variables 1, 2, 3 and 4 are examined.

3.1. First case resolution approach

The objective in the first case is to obtain an optimal loading and unloading pattern when the sequences of incoming and outgoing semi-trailers are known. To do this, we use a graph to demonstrate all possible assignments. The graph nodes and arcs show assignment states and forthcoming possibilities, respectively. The following algorithm, which is based on dynamic programming, is used to construct the graph:

Loading/unloading algorithm (LUA):

Indices:

z: Total incoming semi-trailers

j: Loading destination

l: Total outgoing semi-trailers

N: Node Number

AD: Current loading destination

d: Destinations

Node elements:

$S^N = (H_d, Gd, C, k)$

S^N: Node number N

H_d: Vector of variables indicates possible directly transiting products for each destination

G_d: Vector of variables indicates the number of products in temporary storage for each destination

C: Gain (the total number of direct transiting products from the beginning to the current node)

k: Unloading semi-trailer order number

P_d^k: Number of products for destination d in unloading semi-trailer k

TC: Semi-trailer capacity

Start

Initialize: $H_d=P_d^1, G_d=0, c=0, k=1 \ (\forall d)$ (1)

Create Node: $S^0=(H_d, G_d, c, k)$ (2)

For $j=0$ to l (3)

  For all generated node in loading destination j (4)

    Do until $H_{AD} >= TC+1$ (5)

      If $H_{AD}>=TC$ (6)

        $H_{AD}=H_{AD}-TC, c= c+TC, N=N+1$ (7)

Create Node $S^N=(H_d, G_d, c, k)$ (8)

Restore H_d and c to the values before if statement

Else If $H_{AD}+G_{AD}>= TC$ (9)

$H_{AD}=0, G_{AD}=G_{AD}-(TC-H_{AD}),$ (10)

$c=c+H_{AD}, N=N+1$

Create Node $S^N=(H_d, G_d, c, k)$ (11)

Restore H_d, G_d and c to the values before else statement

If $k<=z$ (12)

$H_{AD}=H_{AD} + P_{AD}^K,\ G_d=G_d+ H_d \quad (\forall d \neq AD), k=k+1$ (13)

Else Exit the loop (14)

Call domination function (15)

END

This algorithm starts with an initial node, (1) and (2). Then, it considers all the generated nodes for the previously loading semi-trailer, (3) and (4), and for each node, it generates all forthcoming nodes for the currently loading semi-trailer: (5) to (14). In (6) to (8) the algorithm considers that the loading products are all directly transferred. In (9) to (11) the algorithm considers storage products in addition to direct products for loading. In (12) to (14) the algorithm unloads the next incoming semi-trailer.

At the end, the node in the last loading semi-trailer that has the highest cost is chosen as the final node. The path from the initial node to the selected final node is the optimal policy for loading and unloading.

In practice, two domination rules (15) are used to omit unnecessary nodes.

Domination Function:

$S^p=(H_d(p), G_d(p), c(p), k(p))$ dominate $S^q=(H_d(q), G_d(q), c(q), k(q))$ if:

1) $K(p) < K(q), H_d(p) = H_d(q)\ \forall d, and\ C(p) \geq C(q)$

2) $K(p) = K(q), C(p) = C(q)\ and \sum_d H_d(q) \leq \sum_d H_d(p)$

The first rule states that, if for each destination, directly transiting products for two nodes are equal, the node with the highest cost dominates the other. The second rule applies when two or more nodes have the same cost, in which case the node with higher summation of directly transiting products for all destinations dominates the others.

Example:

Suppose that there are five incoming and five outgoing semi-trailers with a capacity of ten products each. The outgoing semi-trailers depart to three destinations (2 to destination A, 2 to destination B and 1 to destination C). The outgoing sequence is A-B-A-B-C, and the incoming sequence is I-II-III-IV-V. Table 3-1 presents the products in each trailer; for example, the first unloading trailer has six products for destination A, three for B and one for C.

Table 3-1: Example

Incoming semi-trailer	Semi-trailer contents for each destination		
	A(products)	B(products)	C(products)
I	6	3	1
II	2	7	1
III	5	2	3
IV	3	4	3
V	4	4	2

The loading/unloading algorithm is applied and the generated graph is presented in Figure 3-1. Node 1 is the initial node. The algorithm reads the initial node and generates node 2 with the cost of 10 as a possible assignment for the first loading semi-trailer.

From node 2, nodes 3, 4 and 5 are generated for the second loading semi-trailer. This procedure continues until all outgoing semi-trailers are loaded. Nodes 3 and 5 have the same number of directly transiting products and the cost of node 3 is higher than node 5, but it did not satisfy the domination rule; on the other side, node 10 is dominated by node 9. In the fifth loading semi-trailer, node 12 has the highest cost (29), therefore, it is chosen as the final node and the path with nodes 1-2-3-6-9-12 is the optimal loading/unloading policy.

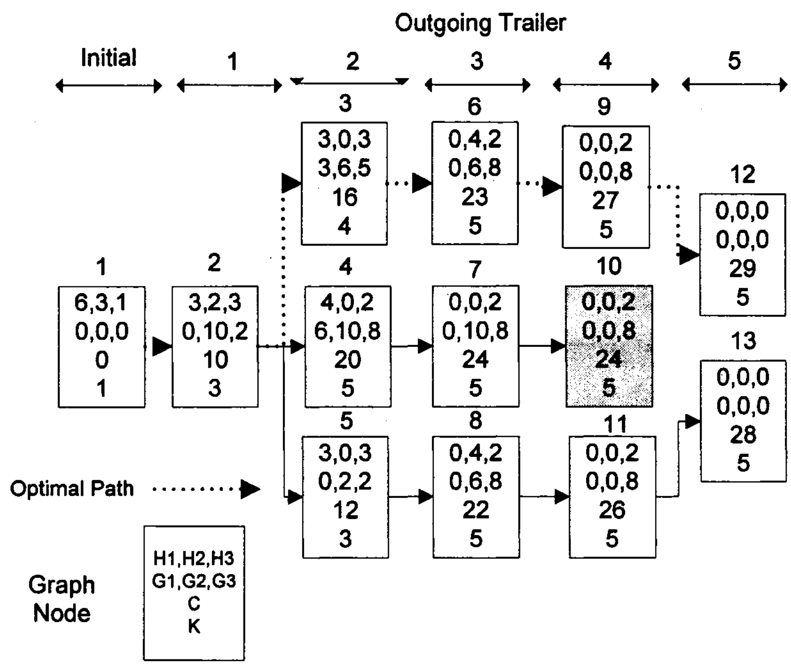

Figure 3-1: Optimal policy algorithm for the example

3.2. Second case resolution approach

In this case, two methods are proposed: Tabu search integrated with the loading/unloading algorithm and a heuristic method. The first method is proposed by the following algorithm.

Sequencing outgoing algorithm (SOA):

Step 1: Run loading/unloading algorithm for the initial value for the number of direct transiting products

Step2: Select two loading semi-trailers' order numbers

Step3: Swap the order numbers and save them in Tabu list (if they are not for the same destination and are not already in Tabu list)

Step 4: Run loading/unloading algorithm

Step 5: Save the cost and, if it is improved, save the optimal path

Step 6: Go to step 2 or stop if the cost is not modified after 20 iterations

In the LUA and SOA, the optimal solution is not the combination of best solutions. In the other words, sometimes selecting the node with the lower cost would lead to the node with the highest cost. For example, node 4 has a higher cost than node 3 However, selecting node 3 leads to the node with the highest cost. In the proposed heuristic method, it is supposed that sub-optimal assignments lead the process to the optimal result.

The proposed algorithm is based on the greedy method, and it assumes that the sequence of incoming semi-trailers is known a priori. In each iteration, the destination that has the highest cost is selected as the outgoing destination. At the end, the selected destinations are the outgoing semi-trailer sequence.

Sequencing outgoing greedy algorithm (SOGA):

Step 1: Create an initial value (the value of directly transiting products for each destination is equal to the first unloaded semi-trailer's products; the rest are zero)

Step 2: For all loading semi-trailers do steps 3 to 8

Step 3: For all destinations do steps 4 to 7

Step 4: Do this as long as total directly transiting product for the selected destination is more than semi-trailer capacity

Step 5: Calculate the cost for current selected destination; preserve the results if improved

Step 6: Consider the next unloaded semi-trailer and update values; go to step 4

Step 7: Save best assignment

Step 8: Set best obtained outgoing assignment as current assignment; go to step 2

Step 9: The final list is the optimal outgoing assignment

Example:

In the previous example, for the given incoming sequence, the solution obtained with the SOA is 34 with the sequence B-A-B-C-A. For the SOGA, the obtained sequence is B-A-C-B-A with the cost of 33.

3.3. Third case resolution approach

The previous methods were developed to obtain the optimal sequence of loading and unloading semi-trailers. In the current method, Tabu search is integrated with the SOA or the SOGA to obtain the best unloading sequence.

Sequencing incoming/outgoing algorithm (SIOA)/ Sequencing incoming/outgoing greedy algorithm (SIOGA):

Step 1: Run SOA (or SOGA) for initial value for the number of direct transiting product

Step2: Select two unloading semi-trailers' order numbers

Step3: If they are not in Tabu list, swap the order numbers and save them in Tabu list

Step 4: Run SOA or SOGA

Step 5: Save the cost and, if it increases, save the optimal path

Step 6: Go to step 2 or stop if the cost is not improved after 20 iterations

Example:

For the example, the cost obtained with the first algorithm is 38 with outgoing sequence B-A-B-C-A and incoming sequence II-I-V-IV-III. For the second algorithm (SIOGA), the outgoing sequence is A-B-A-C-B with I-V-II-III-IV as an incoming sequence and a cost of 37.

4. EXPERIMENTS

In the previous section the resolution approaches for semi-trailer sequencing problems in transhipment platform were studied. We discussed that sequencing semi-trailers is one of the platform operational activities that affect performance. In this section we test the proposed algorithms. We consider small (5 semi-trailers), medium (10 semi-trailers) and large (20 semi-trailers) problems. For small problems, three and five destinations are defined. For medium and large problems, three, five and ten destinations are defined. We assume that the outgoing semi-trailers are equally distributed between destinations. For example, for the medium-sized problem with three destinations, we had four semi-trailers depart to destination I and three semi-trailers depart to destinations II and III, respectively. Table 4-1 presents the distribution of trailers between destinations.
The capacities of incoming and outgoing semi-trailers are 10 units of products. For each defined problem, 20 sets of data are randomly generated.

Table 4-1: Distribution of semi-trailers for each case

		Semi-trailer per destination									
Problem Size	Number of destinations	I	II	III	IV	V	VI	VII	VIII	IX	X
Small	3	2	2	1							
	5	1	1	1	1	1					
Medium	3	4	3	3							
	5	2	2	2	2	2					
	10	1	1	1	1	1	1	1	1	1	1
Large	3	7	7	6							
	5	4	4	4	4						
	10	2	2	2	2	2	2	2	2	2	2

The resolution approaches for the first, second and third cases are tested with generated samples. We completely enumerated all the possible incoming and outgoing sequences for the small problem to obtain the best and the worst bound; we also calculated the average and the standard deviation of each defined problem. Tables 4-1, 4-2 and 4-3 present the results for the small problem.

Table 4-2: The best and the worst bound

	Second case				Third case			
Destination combination	Average	Std	WB*	BB**	Average	Std	WB*	BB**
2-2-1	25.96	4.85	18.2	33.45	25.81	4.93	16.15	37.9
1-1-1-1-1	18.19	1.35	16.6	20.85	18.14	1.31	15.75	23.85

* WB: the average obtained worse bound
** BB: the average of obtained best bound

Table 4-3: Average Number of direct transiting products (small problem) (Max 50 product)

Destination combination	L.U.A	S.O.A	S.O.G.A	S.I.O.A	S.I.O.G.A*
2-2-1	30.4	33.3	32.8	37.4	37
1-1-1-1-1	18.3	20.65	20.55	23.45	23.3

* L.U.A: Loading/unloading algorithm
S.O.A: Sequencing outgoing algorithm
S.O.G.A: Sequencing outgoing greedy algorithm
S.I.O.A: Sequencing incoming/outgoing algorithm
S.I.O.G.A: Sequencing incoming/outgoing greedy algorithm

Table 4-4: Algorithm performance

Second case

Destination combination	WB&BB	S.O.A & L.U.A	S.O.A & BB	S.O.G.A & L.U.A	S.O.G.A & BB
(2-2-1)	30.50%	5.80%	0.30%	4.80%	1.30%
(1-1-1-1-1)	8.50%	4.70%	0.40%	4.50%	0.60%

Third case

	WB&BB	S.I.O.A & L.U.A	S.I.O.A & BB	S.I.O.G.A & L.U.A	S.I.O.G.A & BB
(2-2-1)	43.50%	14.00%	1.00%	13.20%	1.80%
(1-1-1-1-1)	16.20%	10.30%	0.80%	10.00%	1.10%

Table 4-1 shows the best and the worst bound and the average number of directly transiting products. The average number of directly transiting products does not change notably. The small standard deviations indicate that although the average number of directly transiting products in the second case did not differ from the third case, the best and the worst bounds change. The gap between them is increased from 30.50% to 43.50% for 2-2-1 and 8.50% to 16.20% for 1-1-1-1-1 (Table 4-4).

Table 4-2 presents the average number of directly transiting products for each case. The solution for the LUA is an initial answer, which is used as a base to compare the improvements for each case. The improvements of each algorithm (percentàges) are presented in Table 4-4. As can be seen, the improvements of applying greedy algorithms are less than the algorithms based on the LUA, but their gaps with the optimal solution are negligible (0.40% compare to 0.60% for case 1-1-1-1-1). As shown in Table 4-4, the gap between the optimal solution and the algorithms solution is less than 2%, which indicate that in our sample problem both algorithms have good performance. Figures 4-1 and 4-2 illustrate the solutions for each data set in case 2-2-1 of the small problem.

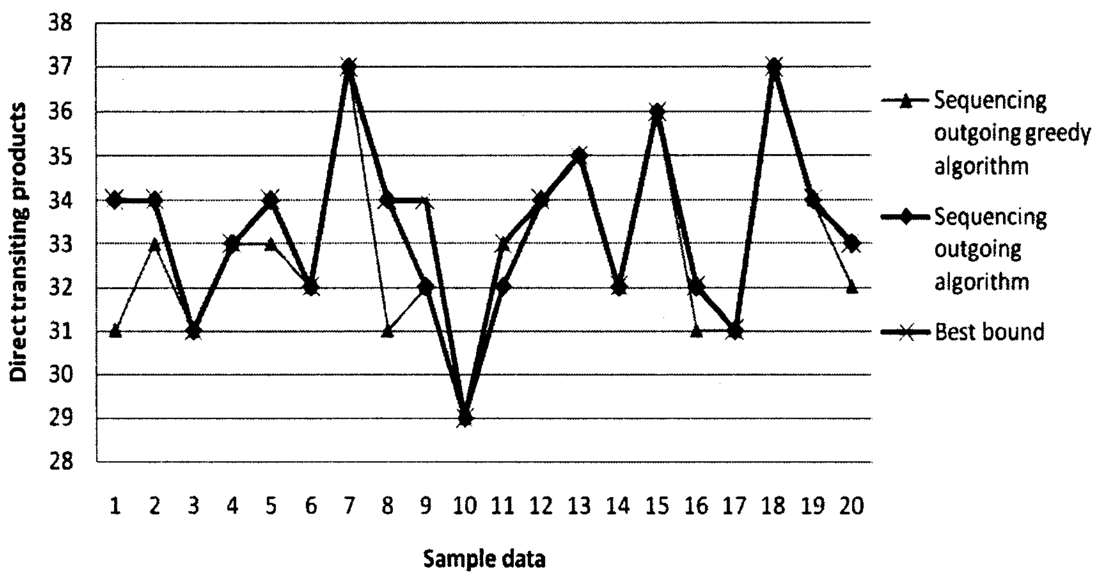

Figure 4-1: Comparing the algorithms' performance for second case with optimal solution (case 2-2-1 data)

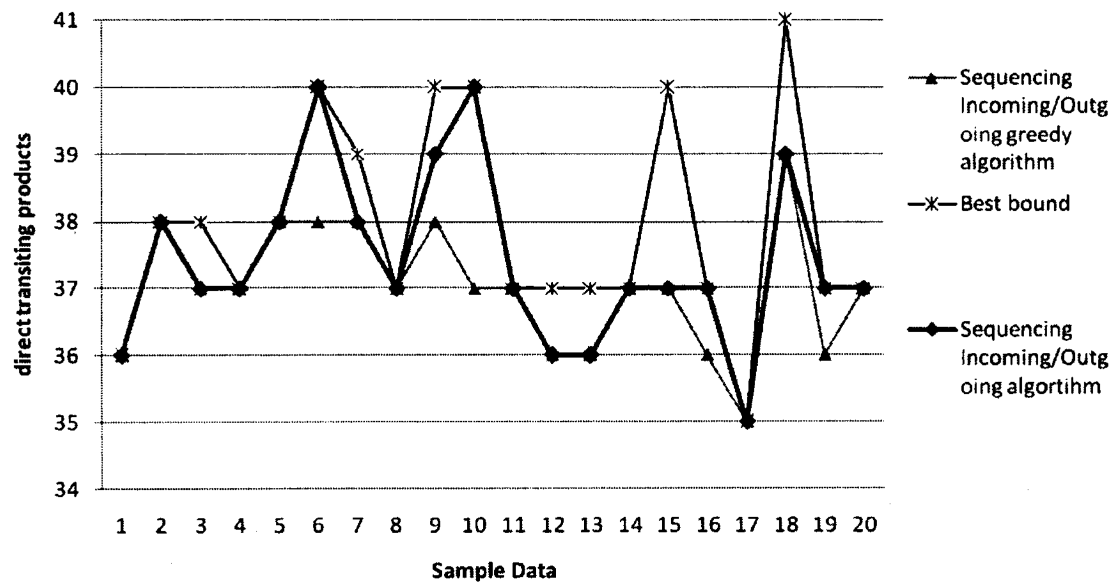

Figure 4-2: Comparing algorithms' performances for third case with optimal solution (case 2-2-1 data)

From Table 4-5, we see that sequencing both incoming and outgoing semi-trailers notably increased the performance. For example the improvement in case 2-2-1 is 8.2% (from 5.80% to 14.00%) for LUA-based algorithms and 8.4% (from 4.80% to 13.20%) for greedy-based algorithms.

The results of the algorithm for medium and large problems are in Table 6. Sequencing both incoming and outgoing semi-trailers notably increases the number of directly transiting products; in fact, it synchronizes the product flow from the receiving to the shipping door. Figures 4-3, 4-4,4-5 illustrate the improvements in the small, medium and large problems. In all cases, sequencing both incoming and outgoing semi-trailers notably increases performance.

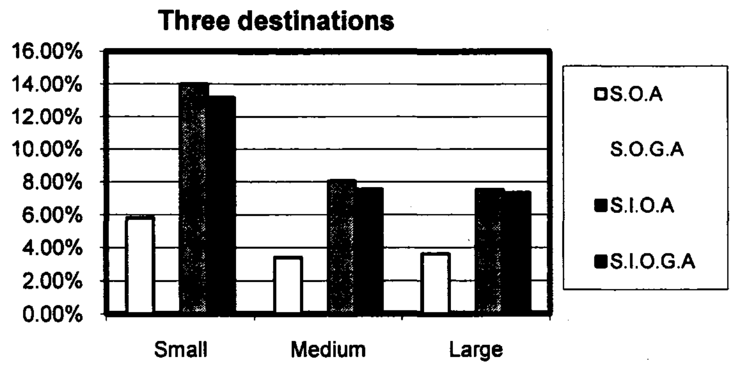

Figure 4-3: Compare algorithms performance for three destinations

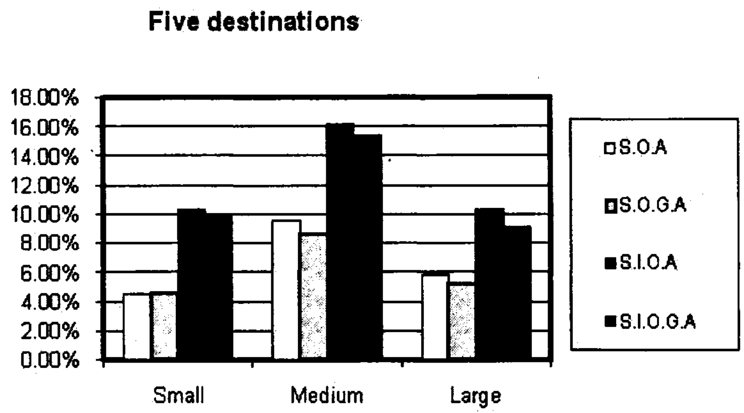

Figure 4-4: Compare algorithms performance for five destinations

Figure 4-5 : Compare algorithms performance for ten destinations

Table 4-5: Algorithms improvements for five,ten and twenty semi-trailers

Problem size	Algorithm	Number of destinations 3	5	10
Small	S.O.A	5.80%	4.50%	N/A
	S.O.G.A	4.80%	4.50%	N/A
	S.I.O.A	14.00%	10.30%	N/A
	S.I.O.G.A	13.20%	10.00%	N/A
Medium	S.O.A	3.40%	9.60%	6.25%
	S.O.G.A	1.80%	8.65%	6.20%
	S.I.O.A	8.05%	16.10%	14.30%
	S.I.O.G.A	7.55%	15.35%	13.50%
Large	S.O.A	3.60%	5.83%	9.25%
	S.O.G.A	2.60%	5.15%	8.15%
	S.I.O.A	7.53%	10.30%	12.28%
	S.I.O.G.A	7.33%	9.10%	12.05%

To conclude, for our generated samples, all of the algorithms noticeably increased performance, but the greedy-based algorithms had fewer improvements than the others. Also, for our small problem, the gap between the optimal and the algorithms solution was less than 2%.

5. CONCLUSION

The transshipment platform is where products from incoming semi-trailers are unloaded and then consolidated with other products for reshipment. The efficiency of such platforms is manipulated by the ratio of directly transiting products to total transferring products.

This research explores the particular case of a platform with a single receiving and a single shipping door. Based on our knowledge about the sequences, three cases of the problem were studied. Dynamic programming (LUA) and a heuristic method (SOGA) were proposed as two major functions to solve the cases. Experiments were performed on small, medium and large problems. In addition, we did complete enumeration for the small problem to investigate the performance of the proposed algorithms.

When both sequences were known, an algorithm based on dynamic programming found the optimal use of the temporary inventory. When incoming sequences were known, two different approaches were proposed. The first used a dynamic programming for optimal loading/unloading and evaluation functions to determine outgoing sequence. For the second, we proposed a fast greedy-based heuristic algorithm. In our generated problems, the gap between the two proposed algorithms is negligible. For the third case, the two previously proposed algorithms were developed with evolutionary search for incoming sequence. Both algorithms present acceptable performance and show notable improvement compared to the second case.

Overall, sequencing incoming and outgoing semi-trailers is one of the operational activities on a cross-docking platform that can increase transshipment performance by increasing the number of directly transiting products. In this work, we assume one inbound and one outbound door. This model should be extended to a general platform with multiple incoming and outgoing doors.

6. REFERENCES

[1] KINNEAR. E., Is there ant Magic in cross-docking. Supply chain management: an international Journal, 2(2):49-52, 1997.

[2] ZHANG. M., Cross-docking network design, PhD thesis in Industrial and system engineering , Georgia institute of technology , 1997

[3] RATLIFF HD., VATE JV, ZHANG M. Network design for load-driven cross-docking systems, http://www.isye.gatech.edu/apps/research-papers/papers/misc9914.pdf

[4] DONALDSON H., JOHNSON EL., RATLIFF HD., ZHANG M, Schedule-driven cross-docking networks, http://www.isye.gatech.edu/apps/research-papers/papers/misc9904.pdf

[5] CHEN P., GUO Y., LIM A., RODRIGUES B. ,Multiple cross docks with inventory

and time windows, Computers and Operations Research 33, 43–63, 2006.

[6] LEE Y.H., JUNG J.W., LEE K.M., Vehicle routing scheduling for cross-docking in the supply chain, Computers and Industrial Engineering, article in press, 2006.

[7] GUE K.R., KANG K., Staging queues in material handling and transportation systems, Proceedings of the 2001 Winter Simulation Conference. 1104-1108,2001.

[8] BARTHOLDI J.J, GUE KR. The best shape for a cross-dock. INFORMS national conference, San Antonio, TX, 2000.

[9] BARTHOLDI J. J., GUE K. R., The Best Shape for a Cross-dock, Transportation Science, 38 (2), 235-244, 2004.

[10] SCHAFFER B., Implementing a Successful Cross-docking Operation, IIE Solutions, 29 (10), 34-37, 1997.

[11] TSUI L. Y., CHANG C. H., Optimal Solution to a Dock Door Assignment Problem, Computers and Industrial Engineering, 23, 283-286, 1992.

[12] TSUI L. Y., CHANG C. H., A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards, Computers and Industrial Engineering, 19, 309-3121990.

[13] LI Y., RODRIGUES B., Cross-docking —JIT scheduling with time windows, Journal of the Operational Research Society 55, 1342–1351, 2004.

[14] MC WILLIAMS D.L., STANFIELD P.M., GEIGER C.D., The parcel hub scheduling problem: A simulation-based solution approach, Computers and Industrial Engineering, 49, 393–412, 2005.

[15] YU W., EGBELU P.J., Scheduling of inbound and outbound trucks in cross-docking system with temporary storage, European journal of operation research, Article in press,2006.

[16] P. BAPTISTE, B. PENZ, R. LARBI, Polynomial time solution methods for some cross-docking problems, International Conference on Industrial Engineering and Systems Management,2007.